Programação para Dispositivos Móveis
Game Store

Imagine que é um programador móvel de sucesso que trabalha em uma grande empresa de jogos. Como todas as empresas de jogos hoje em dia querem que todos continuem a pagar por um jogo mesmo depois de já o terem comprado, foi-lhe dada a tarefa de desenvolver a base para uma nova **aplicação de loja** para os itens dos jogos, expansões e DLCs (Downloadable content) da empresa.

Para criar esta aplicação, devem seguir todas as melhores práticas relacionadas com o desenvolvimento de aplicações móveis, incluindo a organização de diferentes elementos/componentes de UI em classes/ficheiros separados e, especialmente, o padrão de design MVC (Model-View-Controller).

Como foi encarregado de criar apenas a base da aplicação, a mesma não deve executar qualquer pedido à web para este projeto, embora a aplicação deva ser desenvolvida com o fato de que haverá um backend no futuro e todas as providências para tal necessitam serem feitas.

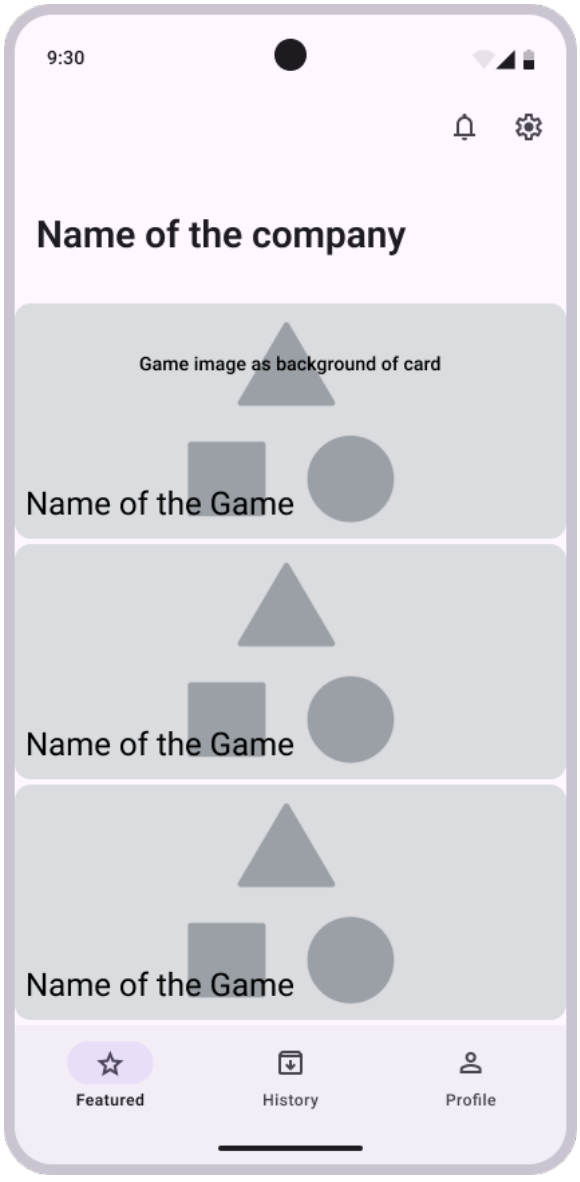
A aplicação deve ser desenvolvida utilizando Kotlin e Jetpack Compose e não deve utilizar qualquer biblioteca externa para além das que se encontram no modelo base de projeto do Android Studio. Utilizar bibliotecas de terceiros anulará a tarefa por completo.

Será necessário desenvolver duas Activities (denominadas `MainActivity` e `GameDetailActivity`), o utilizador deve ser capaz de navegar entre elas (para a frente e para trás) sem a utilização de `NavHosts`, apenas `Intents`, e devem ser passados objetos de modelos entre elas (`Activity.putExtra`) para enviar informação a ser exibida.

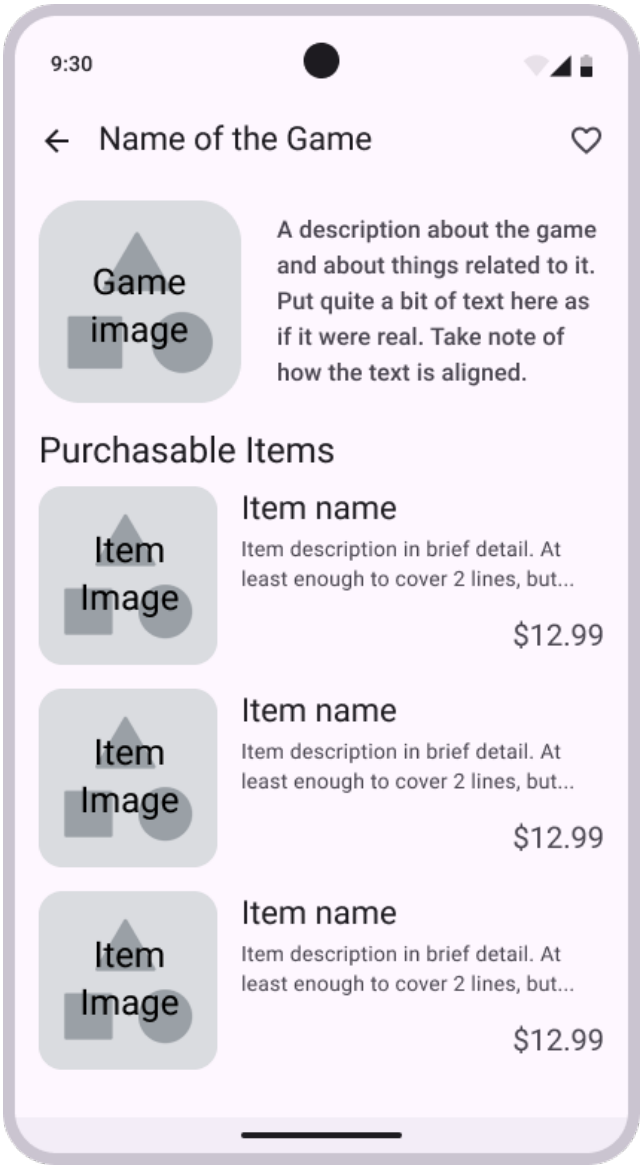
A fim de apresentar ao utilizador uma amostra do funcionamento da aplicação, deverão haver modelos de objetos de dados de exemplo. Os modelos criados para o projeto devem seguir todas as melhores práticas da indústria, assim como demonstrado em sala e devem conter toda a informação para a

demonstração. Estes objetos devem ser instanciados uma vez na `MainActivity` e devem imitar a forma como trabalharia com eles no seu código como se viessem de um serviço web.

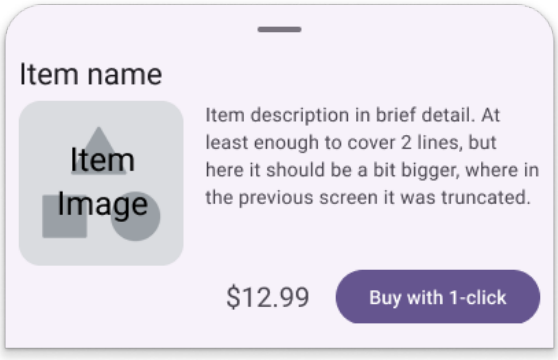
A `MainActivity` deve ter uma interface que seja exatamente como se segue:



Esta `Activity` deve permitir ao utilizador selecionar um jogo a partir de uma lista de `Cards` de jogo e navegar para a `GameDetailActivity`, a qual deve ter o seguinte layout:



Nesta `Activity`, sempre que o utilizador selecionar um item comprável da lista rolável, deve aparecer uma `ModalBottomSheet` com o seguinte layout:



Ao carregar no botão para comprar, a `BottomSheet` é encerrada e deverá ser apresentado um `Toast` com a seguinte mensagem “Acabou de comprar o item X por $Y”, onde X é o nome do item que foi comprado e o Y é o preço do mesmo.

Para que a aplicação de exemplo pareça real, os dados de exemplo criados para ela devem conter, no mínimo, 2 jogos com pelo menos 3 itens compráveis cada. Os objetos compráveis devem ser diferentes entre os dois jogos. Todas as informações contidas nos exemplos podem ser copiadas da Internet e não necessitam serem exatas, apenas representativas para efeitos de exemplo.

Todos os elementos de UI criados devem ter `Composable Previews` associadas, seguindo as melhores práticas da indústria, assim como demonstrado em sala.

Quaisquer recursos, como imagens ou ícones, necessários para a aplicação, sejam eles elementos da interface do utilizador ou dados de exemplo, devem ser descarregados e importados para o projeto de forma adequada.

Deverá submeter os seguintes materiais para a avaliação:

- **Ficheiro ZIP do projeto completo do Android Studio**
  - Projetos com ficheiros em falta (classes ou ficheiros de configuração) ou corrompidos anularão toda a submissão.
- **Link para o repositório do projeto no GitHub**
  - Projetos sem link do repositório ou com código diferente do que foi submetido via ZIP terão a tarefa anulada.
- **Ficheiro APK com a build do projeto para correr no emulador**
- **Screenshots do Android Studio,** por completo, a mostrar uma preview (ou screenshot do emulador) de cada Activity criada para este projeto.
  - Estes screenshots serão também utilizados para verificar o seu trabalho, pelo que a não apresentação dos mesmos resultará também em uma reprovação na tarefa.

O projeto é avaliado com base em duas componentes: quantitativa (A), e qualitativa (B). A nota final do projeto é determinada por (0.5 x A) + (0.5 x B).

A avaliação quantitativa provém da implementação estrita de todas as instruções descritas neste enunciado.

A avaliação qualitativa irá considerar que existem várias formas de resolver o problema descrito, mas exige-se a utilização dos instrumentos e métodos apresentados na unidade curricular, nomeadamente:

- Separação entre interface, dados e lógica da aplicação
- Justificação clara para as variáveis e operações implementadas, além dos métodos (funções) utilizados
- Organização da solução coerente com a metodologia apresentada na unidade curricular
- Adequação da escolha de estruturas de dados e algoritmos para a resolução do problema

Todos os projetos entregues serão sujeitos a prova de autoria. Para esse efeito, terá de efetuar uma discussão com a docência, de forma a demonstrar que o código entregue foi de facto feito por si. A não comparência na prova de autoria implica a anulação da componente qualitativa.

A **tarefa deverá ser entregue até o dia 05/12 às 23:59**. Esta data e horário são absolutamente inflexíveis e qualquer trabalho submetido após esta deadline será desconsiderado e a nota será anulada.

A **prova de autoria será realizada na semana do 08/12**, ou em uma data acordada com o docente.

Quaisquer dúvidas relacionadas com ambiguidades encontradas neste briefing deverão ser questionadas e esclarecidas durante o período de aula.

Vale ressaltar que quaisquer situações de plágio, seja entre alunos ou por utilização indevida de fontes da internet ou ferramentas IA, resultará na anulação total da tarefa.