MercadoLibre - Examen Mobile Developers

Es **importante** que las respuestas las escribas con tus palabras. Si consultás alguna fuente de información porque desconoces un tema, te pedimos que por favor nos indiques cuáles son.

Conceptos generales de programación

1. Definir que es una Clase y que es una instancia de una clase

R: Classe é a representação de um objeto e/ou de uma receita de um objeto na qual possui os atributos e comportamentos do objeto, já a instancia da classe, é quando colocamos ela na memória e podemos executar os comportamentos e ler o valores gravados nas suas variáveis, que são os atributos.

2. Explique los siguientes conceptos:
    a. Herencia: **R: É uma forma de compartilhar atributos de uma classe pai para as suas classes filhas, podendo assim também compartilhar métodos, por sua vez, no java não existem heranças múltiplas, uma classe "filha" só pode ter uma classe "pai", mas esta classe "pai" pode ser herança de N filhas.**

    b. Polimorfismo: **R: É uma forma de definir objetos, atributos e métodos para serem usados em toda a aplicação, na qual eles podem ter comportamentos diferente em determinadas classe.**

    c. Encapsulamiento: **R: É uma forma de "esconder" atributos e métodos de um classe ou objeto para outro objeto, geralmente usando os modificadores de acesso.**

3. ¿Qué es un patrón de diseño?. Ejemplificar con patrones usados en la plataforma.

R: São padrões consolidados e testados pela comunidade, estes foram definidos e descritos pela GOF, na qual trás processos e formas de desenvolvimento que facilite a manutenção e der confiança ao projeto, um dos exemplos muito usados no Android, é também um dos mais conhecidos, o Singleton na qual consiste em instanciar um objeto que será usado várias vezes durante a execução do app e não deixar ele ser executado novamente por outra classe, conseguindo manter a mesma instancia para todo o app.

4. ¿Qué es una interfaz?. Compare la misma con una clase abstracta

R: Uma interface, basicamente é um controle, na qual serve para especificar de forma abstrata os comportamentos da classe que ela será implementada. A grande diferença entre classe abstrata e interface, é que a classe abstrata pode ter métodos incompletos, como a interface, e pode ter métodos completos que as interfaces por sua vez não podem, este é o diferencial mais marcante.

5. ¿Qué es un Thread? Describa brevemente qué consideraciones hay que tener cuando 2 threads acceden a un mismo recurso

R: Thread são classes que permitem executar métodos e tarefas sem ocupar a Activity principal, isso é crucial no android, pois se o processo for demorado e ocupar a Activity o Android para a aplicação, então a Thread se encarrega de executar esse processo fora da Activity principal por exemplo. Executando em paralelo de modo Assíncrono. No caso de risco das Threads necessitarem usar o mesmo recurso, no seu interior elas devem ser implementadas de forma Síncronas e até mesmo uma fila, ainda estará executando fora da Thread da Activity Principal mas esperará a Thread que está acessando o recurso terminar o processo.

6. ¿Qué es una Exception?, ¿cómo se puede manejar?

R:**Exceptions são basicamente erros e ocorrências anormais que pode afetar o funcionamento da aplicação, então elas ajudam quando são para não deixar a aplicação parar e conseguir contorná-las, como o próprio nome já diz, não é um erro, é uma excessão a ser tratada, para evitar transtorno ao usuário.**

Programación y Java (puede haber más de una rta correcta)

1. equals() y hashCode()
    a. Son equivalentes
    b. No tienen relación
    c. Dos objetos iguales para equals deben tener el mismo hashcode **- CORRETO**
    d. Dos objetos distintos pueden tener el mismo hashcode **- CORRETO**
    e. Son fundamentales para meter objetos en un hashSet
2. Synchronized...
    a. Es para evitar que más de un hilo ejecute una sección del código al mismo tiempo **- CORRETA**
    b. Es una palabra reservada en Java **- CORRETA**
    c. Se puede usar como modificador de una variable de instancia
    d. Se puede usar como modificador de un método de instancia
    e. Se puede usar como modificador de una clase
    f. Se puede usar para sincronizar sólo algunas líneas dentro de un método.

3. HTTP (HyperText Transfer Protocol)
    a. Es un programa para navegar internet
    b. Es un protocolo stateless **- CORRETO**
    c. Para un request solo corresponde un response **- CORRETO**
    d. Permite transferir únicamente Hipertexto.
    e. GET y POST Son equivalentes
    f. HTTP siempre viaja encriptado para evitar que alguien pueda ver el contenido.
4. Explique qué entiende por REST y de un ejemplo

R: Rest é uma arquitetura para aplicações WEB mais especificamente para APIs e ele tem por objetivo focar no papel dos componente dispensando a implementação de sintaxe de protocolo, e é a forma mais simples de se escrever e consumir uma API hoje em dias, um exemplo bem claro é as requisições GET que podem ser enviadas Query Params e sua resposta pode ser de n formas, JSON, Texto, até mesmo uma renderização de página.

5. Enumere tres headers de HTTP y su funcionalidad

R: -Authorization: Este header vai a credencial de autenticação do usuários, normalmente um token.

-Accept: Informa o tipo dos dados suportados e que podem retornarem na resposta.

-Content-Type: Indica o tipo de informação(body) ou mídia enviada para o servidor.

6. ¿Qué son y qué diferencias hay entre un Thread (Hilo) y un Proceso ?
    a. Son sinónimos **- CORRETO**
    b. Un proceso puede tener múltiples threads **- CORRETO**
    c. Un thread puede tener múltiples procesos
    d. Los procesos no comparten espacio de memoria **- CORRETO**
    e. Los threads no comparten espacio de memoria
7. ¿Cómo se logra la ejecución de más de 1 hilo en un procesador con un solo core (núcleo)?
    a. No es posible
    b. Se particiona el core en partes para cada hilo
    c. Se turnan los hilos para correr en el mismo core en espacios de tiempo pequeños. **- CORRETO**
    d. Se turnan entrando uno al terminar el trabajo del anterior. **- CORRETO**
    e. Se generan nuevos cores a demanda, para cada hilo.

Conocimientos de Android

1. ¿Qué es una Activity? Definir su ciclo de vida.

R: Uma activity é basicamente uma classe responsável por gerenciar toda a UI. Seu ciclo de vida inicia o onCreate(Que é responsável por carregar todo o XML do UI e esse executa apenas uma vez**), vai para o onStart(**É chamado também quando a activity está em background e volta ao foto por exemplo**), em seguida pro onResume(**este é semelhante ao onStart, mas o onStart so é chamado quando a activity não era mais visível, ja o onResume é chamado a cada vez que volta o foco, geralmente quando a onPause entra em ação, logo em seguida quando volta o foco passa por aqui**) e só aí a activity inicia de fato, onDestroy(**Esta é a ultima a ser executada, quando a instancia da Activity é destruída por exemplo**) e ainda temos a onRestart(**Que é chamada antes da onStart quando a activity volta do background**).**

2. ¿Puedo llamar a una Activity y esperar un resultado?

R: Não, uma activity nunca retorna um resultado.

3. ¿Qué es un Fragment? Explicar las diferencias con respecto a un Activity.

R: Fragmento pode ser todo ou apenas uma parte da representação do comportamento do usuário, como por exemplo uma activity pode conter vários fragments que juntos representam todo o comportamento do usuário. A principal diferença é que a Activity sempre que chamada ela visualmente é única, e o fragmente pode ser chamado por várias activityes e locais diferentes como também pode ser montados vários fragments dentro de uma única activity.

4. ¿Cómo se comunican un Fragment con una activity? ¿Es posible la comunicación en ambos sentidos?

R: Sim, é possível pois o fragmento acessa a activity com o método nativo getActivity fazendo um casting do mesmo para a activity referida, mas ele precisa acessar alguma variável ou método publico na Activity para conseguir comunicar, ja da activity para o fragment, pode ser usado o Bundle, ou até mesmo uma variável ou método publico no fragment, pois a activity tem uma instancia acessível dele.

5. ¿Qué son los recursos y cómo funcionan?

R: Recursos são libs que já veem no SDK do Android para facilitar a implementação de funcionalidades que usem recursos do SO e recursos físicos, como a lib de GPS, a de Câmera, entre outras, na qual o desenvolvedor não tem necessidade de recorrer a uma lib de terceiro.

6. Qué es y para que se utiliza: AndroidManifest, Intent, Service, AsyncTask, Handler, LayoutInflater, Adapter y Gradle.

R: -AndroidManifes: Basicamente define o esqueleto da aplicação, as

permissões, as activityes, os services, etc… Ele é responsável também por definir como a aplicação se comportará como rotações de tela, abertura de teclado entre outros.

-Intent: Basicamente é uma classe que permite coordenar funções para executar tarfeas, um exemplo é quando precisa abrir uma nova activity, é o intent que entra em ação.

-Service: basicamente é um serviço na qual sempre está rodando em background executando tarefas automáticas para melhorar a experiencia do usuário. Um exemplo são as notificações e monitoramento de geolocalização que usam constantemente os services.

-AsynTask: É responsável por executar tarefas Assíncronas utilizando uma thread, a grande vantagem dele é que disponibiliza métodos com um ciclo de vida bem consistente possibilitando demonstrar até mesmo o progresso em detalhes da tarefa que está sendo executada para o usuário.

-Handler: Ela se assemelha a uma thread porém, diferente da thread ela consegue acessar a thread principal, na qual é responsável pela UI e atualizar componentes.

-LayoutInflater: Como o próprio nome sugere, é um componente responsável por "inflar layouts" geralmente em componentes nativos que é necessário implementar um customizado, como por exemplo um menu do ActionBar, um conteúdo customizado no Dialog entre outros.

- Adapter: É responsável por manipular a renderização de cada item de uma lista como um ListView ou RecyclerView, até mesmo de um fragment com abas por exemplo.

-Gradle: É responsável pela compilação do projeto, pois nesse é controlado todas as Libs que são adicionadas, até mesmo parâmetros de compilação, versão do app e seguranças adicionais como a ativação do Proguard, basicamente é um sistema que autonomiza a compilação do código e das libs.

Patrones de diseño Android

1. ¿Qué es el ActionBar y qué uso puedo darle?

R: A ActionBar é a barra superior do aplicativo, onde geralmente fica o título da janela, alguns botões de navegação e alguns atalhos, como o próprio nome sugere, é uma barra de ações (Rápidas), que substituiu gradativamente o ToolBar.

2. ¿Qué diferentes tipos de navegación se utilizan regularmente en Android? Enumerarlas y definirlas.

R: - Drawer Menu: Se trata do menu do lado esquerdo da aplicação, na qual é encontrado todas as possíveis rotas primárias, e ele pode ser acessado tanto em um ícone na ActionBar como com um gesto de arrastar da esquerda para a direita na tela.

-FAB(Floating Button): Se trata de um menu flutuante que sempre está visível

ao usuário e em alguns casos pode até ter sua própria árvore de opções.

-TabBar: Se trata de uma barra superior(mais usado) ou inferior na qual é colocado as principais rotas do app ou da sessão que está sendo acessada, que geralmente ficam entre 3 a 5 ítens, para fácil entendimento so usuário.

3. ¿Que es el Android Support Library?. ¿Qué beneficios podemos obtener al utilizarlo?

R: Se trata do conjunto de manuais(guias) na qual foram desenvolvidos para auxiliar o uso e implantação de Bibliotecas produzidas pela equipe desenvolvedora do Android, a vantagem é que por se tratar da equipe responsável pela programação do sistema, as dicas contidas nele por si só tem um ar de "maior confiança" e com isso conseguimos(devs) trazer recursos com maior estabilidade.

Ejercicios de código

Ejercicio A

Dada la siguiente Activity, explique las líneas de código numeradas:

```
public class EjercicioActivity extends Activity {
        @Override
        public boolean onCreateOptionsMenu(Menu menu) {                          //1
                getMenuInflater().inflate(R.menu.menu_ejemplo);                  //2
                return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {            //3
                switch (item.getItemId()) {
                        case android.R.id.home:                                  //4
                                // TODO
                                return true;
                        default:
                                return super.onOptionsItemSelected(item);
                }
        }
}
```

Línea 1) ¿Para qué se utliiza dicho método?
R: Este método é utilizado quando se quer criar um menu especifico e voltado a aquela activity.
Línea 2) ¿Qué realiza dicha línea de código?
R: Infla o layout de menu "menu_ejemplo" como menu de contexto daquela Activity.
Línea 3) ¿Cuál es el propósito de dicho método?
R: Ele é acionado a cada toque nos ítens do menu inflado anteriormente.
Línea 4) ¿Qué se recomienda hacer dentro de dicho Case?
R: Deve executar as rotinas pertinentes ao clique no componente "Home"do menu, ao que parece, desde ser chamado a tela principal, que se por sua vez for a primeira antes desta Activity,um simples `this.finish();`.

Ejercicio B

Definir qué realiza el siguiente código:

```
String nombre = "Roberto";
Intent intent = new Intent(this, CualquierActivity.class);
intent.putExtra("nombre", nombre);
startActivity(intent);
```

R: Descrevendo linha a linha, a linha número 1 cria uma variável chamada nombre com o valor "Roberto", a linha número 2 cria uma instancia do Intent para futuramente chamar a Activity 'CualquierActivity', a linha número 3 envia para o intento o valor da variável nombro para um Extra que será recebido na Activity ao ser

acionada, a linha 4 finalmente chama a activity que foi preparada pela linha 2 e ela recebe o valor da variável nombre através do Bundle com seguinte código: *getIntent().getExtras().getString("nombre");*.

Ejercicio C (mayor importancia)

Con este ejercicio se busca que muestres conocimientos a la hora de crear un proyecto en Android, ver si usas librerías y cómo planteas la solución. **Es importante que cuando respondas el cuestionario, la respuesta a este ejercicio sea un link a Github, Bitbucket o algún repositorio de código.**

MercadoPago posee APIs abiertas a la comunidad para que cualquier desarrollador las consuma y pueda tener pagos en su aplicación. Pero las APIs necesitan de una clave pública para identificar quién es el que está invocando a la API, además de ofrecer customizaciones para esa persona en particular. En este ejercicio, vas a utilizar la siguiente public_key: **444a9ef5-8a6b-429f-abdf-587639155d88**.

A la hora de realizar un pago con tarjeta de crédito, se debe especificar el monto, el medio de pago, el banco y la cantidad de cuotas, además de una identificación segura (token) de la tarjeta con la que se está realizando el pago.

Te pedimos que realices una secuencia de pantallas realizando distintos API Calls para obtener todos los datos (exceptuando el token). El input de la pantalla actual son todos los seleccionados por el usuario en las pantallas anteriores.

La secuencia de pantalla debe ser:

1. **Pantalla de ingreso de monto.**
2. **Selección de medio de pago** (de tipo **credit_card**). Los medios de pago puede ser: Visa, Mastercard, American Express, etc. **Mostrar nombre e imagen.** Para acceder a estos medios de pago, se consulta a la siguiente URL: https://api.mercadopago.com/v1/payment_methods?public_key=PUBLIC_KEY (Método GET).
3. **Selección de banco**. Los bancos pueden ser: ICBC, Santander, Galicia, entre otros. **Mostrar nombre e imagen.** Estos operan con diversos medios de pago de pago, por lo que debes consumir una API para poder saber cuáles son los bancos disponibles para el medio de pago seleccionado. https://api.mercadopago.com/v1/payment_methods/card_issuers?public_key=PUBLIC_KEY&payment_method_id=MEDIO_DE_PAGO_SELECCIONADO (Método GET).
4. **Selección de cuotas**. Luego de tener el medio de pago y las el banco seleccionados, el usuario debe seleccionar la cantidad de cuotas en las que desea pagar el monto ingresado en el paso 1. La API provee un ***recommended_message*** que resuelve el mensaje que se debe mostrar al usuario con la cantidad de cuotas, el valor de cada cuota y el monto final. **Mostrar el recommended_message.** https://api.mercadopago.com/v1/payment_methods/installments?public_key=PUBLIC_KEY&amount=MONTO_DEL_PASO_1&payment_method_id=MEDIO_DE_PAGO_SELECCIONADO&issuer.id=ISSUER_SELECCIONADO (Método GET).

Utilizar un esquema de callbacks para cada servicio utilizado. Modularizar la llamada para recibir base_url, uri y query_params.
El ejercicio debe volver a la primer pantalla y mostrar un mensaje de alerta con los valores seleccionados en el flujo de pantallas solicitado.
No es necesario realizar **todo** el modelo de clases, pero se valorará la calidad de la solución entregada y la profundidad del tema.
Recordá entregar el proyecto en un **repositorio**.
La resolución del ejercicio es totalmente libre, todo lo que hagas de más va a sumar en tu proceso de selección.

Buena suerte y gracias por tu tiempo!