**Aplicações Móveis**

Exercícios de Revisão

**Coordenação de Engenharia Informática**

Departamento de Engenharias e Tecnologias

Instituto Superior Politécnico de Tecnologias e Ciências

Nome : Marcelo Rocha - 20210032

# Replicação

1. **O modelo de replicação difere do cache do lado do dispositivo de várias maneiras importantes. Cite duas e explique seu significado.**

**R:**

* **Cópia de Coleções Inteiras x Armazenamento Sob Demanda:** Na replicação, uma coleção inteira ou parcial de dados é copiada para o dispositivo em um único momento e é atualizada periodicamente por meio de um protocolo de sincronização. Por outro lado, no modelo de cache, os dados são buscados sob demanda, ou seja, somente quando o usuário tenta acessá-los.
* **Significado:** Esse comportamento permite que o modelo de replicação garanta acesso a uma coleção mais ampla de dados, mesmo sem conectividade contínua. Em contraste, no modelo de cache, dados não previamente acessados podem não estar disponíveis, o que pode causar falhas de acesso caso o servidor não esteja acessível.
* **Impacto na Exclusão de Objetos:** Quando um objeto replicado é excluído em um dispositivo, ele também é removido da coleção de dados no site mestre e em todas as outras réplicas. No modelo de cache, a exclusão geralmente significa apenas que o objeto é descartado do armazenamento local do dispositivo, mas permanece disponível no servidor mestre.
* **Significado:** No modelo de replicação, a exclusão tem implicações globais, afetando todos os dispositivos e o site mestre, enquanto no modelo de cache, ela é isolada ao dispositivo específico, permitindo maior flexibilidade, mas sem coordenação global dos dados.
* **Falha ao Ler Dados Não Residentes:** No modelo de replicação, se um dado não estiver presente no dispositivo replicado, a tentativa de acesso falha imediatamente, sem buscar o dado no servidor mestre. No modelo de cache, no entanto, uma falta de cache (cache miss) resulta em uma solicitação ao servidor mestre para buscar o dado e armazená-lo no cache para acessos futuros.
* **Significado:** Esse comportamento no modelo de replicação torna os dados replicados totalmente dependentes de sincronizações periódicas. Caso um dado não esteja presente no dispositivo, ele será inacessível até que uma nova sincronização ocorra. No modelo de cache, o dispositivo tem maior flexibilidade para buscar dados sob demanda, o que reduz a probabilidade de falhas de acesso, mas pode aumentar a dependência da conectividade com o servidor.
* **Adição Implícita de Objetos:** No modelo de replicação, novos objetos que se tornam parte da coleção replicada são automaticamente adicionados à réplica no dispositivo durante a sincronização. No modelo de cache, objetos só são adicionados ao dispositivo quando são explicitamente acessados pelo usuário.
* **Significado:** O modelo de replicação garante que o dispositivo tenha acesso a uma coleção completa ou parcial atualizada automaticamente, o que é útil para trabalhar offline. Já no modelo de cache, a adição de novos dados depende diretamente das ações do usuário, o que pode levar a lacunas no armazenamento local caso objetos importantes não sejam acessados previamente.

1. **Dê alguns exemplos de aplicações que ilustram o uso da replicação.**

**R:**

* **Aplicativos de Música Offline (como Spotify e Apple Music):** Permitem que os usuários baixem playlists ou álbuns inteiros para reprodução offline. As músicas são replicadas no dispositivo, permitindo acesso mesmo sem conectividade com a internet.
* **Aplicativos de Mapas Offline (como Google Maps e HERE WeGo):** Oferecem a opção de baixar mapas inteiros para navegação offline. O mapa replicado no dispositivo permite navegação e pesquisa de locais sem necessidade de conexão à internet.
* **Aplicativos de Leitura Offline (como Kindle e Google Play Livros):** Permitem baixar livros ou artigos para leitura offline.Uma cópia do livro ou artigo é replicada localmente no dispositivo, garantindo acesso mesmo sem conexão.

1. **Compare as abordagens “Acesso Remoto a Dados” e “Replicação mestre”.**

**R:**

O acesso remoto a dados e a replicação mestre são abordagens distintas para gerenciar dados em dispositivos móveis, com diferenças significativas em conectividade, consistência e desempenho. No acesso remoto a dados, os dispositivos acessam informações diretamente de um servidor central, o que garante consistência forte, já que os dados acessados estão sempre atualizados. No entanto, essa abordagem depende de uma conexão de rede contínua, o que pode limitar a acessibilidade em caso de indisponibilidade do servidor ou alta latência em redes móveis ou satelitais. Além disso, o consumo de energia é maior devido à comunicação constante com o servidor. Por outro lado, a replicação mestre armazena cópias completas ou parciais dos dados localmente nos dispositivos, proporcionando acesso rápido e offline, ideal para cenários de conectividade intermitente. Contudo, essa abordagem pode apresentar consistência eventual, pois as atualizações feitas localmente são sincronizadas com o servidor mestre apenas quando possível, podendo surgir conflitos de atualização que precisam ser resolvidos. Enquanto o acesso remoto reduz a necessidade de armazenamento local e é mais adequado para cenários em que a conexão estável é garantida, a replicação mestre exige mais espaço no dispositivo, mas oferece maior autonomia e acesso local imediato. Ambas as abordagens têm aplicações específicas: o acesso remoto é amplamente usado em navegação na web e consultas a bancos de dados corporativos, enquanto a replicação mestre é comum em sincronização de e-mails e reprodução offline de conteúdo.

****

1. **Em um modelo de replicação mestre, qual componente é responsável por detectar quando dois dispositivos produzem actualizações conflituantes? Todos esses conflitos podem ser resolvidos automaticamente?**

**R:**

No modelo de replicação mestre, o servidor mestre é o componente responsável por detectar quando dois dispositivos produzem atualizações conflituantes. Embora o servidor mestre possa, em alguns casos, resolver automaticamente esses conflitos, nem todos os conflitos podem ser resolvidos dessa forma. Alguns conflitos podem exigir a intervenção humana para serem resolvidos, especialmente quando as atualizações feitas em diferentes dispositivos violam invariantes ou regras de consistência dos dados que não podem ser automaticamente reconciliadas.

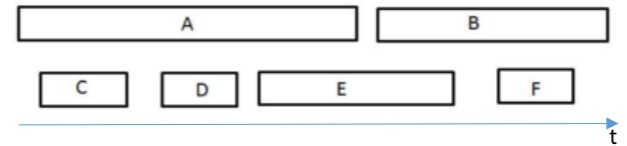
1. **Como compara as diversas soluções de acesso a dados (acesso remoto, dispositivocaching, replicação mestre, etc.) em relação a dispositivos, dados, leituras e actualizações?**

**R:**

****

1. **A distância semântica entre ficheiros pode ser calculada de três formas: i) distância semântica temporal, ii) distância semântica baseada em sequência e iii) distância semântica de tempo de vida (*lifetime*).**

* 1. **Qual é a noção de cada uma?**
  2. **Cite uma desvantagem para cada uma das duas primeiras.**
  3. **Utilizando a terceira (*lifetime semantic distance*), quais são as distâncias entre todos os ficheiros indicados na figura abaixo?**

****

**Figure 1:** **Linha de tempo das operações "abrir" e "fechar" nos ficheiros A, B, C, D, E e F.**

**R:**

* **Distância Semântica Temporal (Temporal Semantic Distance):** A distância semântica temporal entre duas referências de arquivos é igual ao tempo de relógio decorrido entre as referências. Esta definição reflete a ideia de que arquivos referenciados no mesmo momento tendem a estar semanticamente relacionados. No entanto, essa definição pode ser problemática, pois a noção de "tempo" em computadores pode não coincidir com a percepção humana, podendo ser distorcida por fatores como interrupções ou variações na carga do sistema. Além disso, a distância temporal pode ser afetada por anomalias, como longos períodos de inatividade entre acessos a arquivos relacionados, como ocorre ao editar arquivos de código-fonte de um programa ao longo do tempo.
* **Distância Semântica Baseada em Sequência (Sequence-based Semantic Distance):** A distância semântica baseada em sequência entre duas referências de arquivos é igual ao número de referências intermediárias a outros arquivos. Esse conceito tenta medir a proximidade semântica com base na sequência de acessos aos arquivos, sem levar em conta o tempo exato em que ocorreram. A ideia é que arquivos acessados em sequência (um após o outro) são mais prováveis de estarem relacionados semanticamente, como no caso de arquivos de um mesmo projeto. Essa abordagem evita as distorções causadas pelos problemas de tempo descritos anteriormente, usando a sequência de referências para capturar a relação entre arquivos. Essa forma de medição também pode ser mais eficiente, já que se concentra apenas na ordem de acesso e não nas horas exatas. A desvantagem é que ela pode não capturar adequadamente as relações semânticas em situações de trabalho intenso em um único arquivo. Por exemplo, se um arquivo for acessado repetidamente sem referência a outros arquivos entre as aberturas, a distância semântica baseada em sequência será maior do que a relação real entre os arquivos. Isso ocorre porque a abordagem considera apenas a sequência de referências, e não o contexto em que essas referências ocorrem. Além disso, se a sequência de referências incluir repetições de um mesmo arquivo, a abordagem pode resultar em distâncias semânticas que não refletem com precisão a real proximidade semântica entre os arquivos, já que cada repetição é contada como uma nova referência, podendo exagerar a distância entre arquivos que estão, de fato, fortemente relacionados.
* **A Distância Semântica de Tempo de Vida (Lifetime Semantic Distance)**: A distância semântica de tempo de vida entre a abertura de um arquivo A e a abertura de um arquivo B é definida como 0 se A não foi fechado antes de B ser aberto, e o número de aberturas de arquivos intermediários (incluindo a abertura de B) caso contrário. Esta definição leva em consideração o ciclo de vida de um arquivo, ou seja, o tempo entre sua abertura e fechamento. A distância semântica de tempo de vida é particularmente útil para capturar relacionamentos de longo prazo entre arquivos que são manipulados em conjunto, como ao compilar um programa em que diferentes arquivos são abertos e fechados em sequência. Por exemplo, se um arquivo fonte permanece aberto durante todo o processo de compilação, enquanto os arquivos de cabeçalho são abertos e fechados em sequência, o relacionamento entre o arquivo fonte e os arquivos de cabeçalho seria considerado mais próximo, com distância semântica de tempo de vida igual a 0.

**c )** supondo que os expoentes 'o' e 'c' indicam aberturas e fechamentos, respectivamente , as distâncias são: --> = 0 , --> = 0 , --> = 0 , --> = 4 , --> = 5 , --> = 1 , --> = 2 , --> = 3 , --> = 4 , --> = 1 , --> = 2 , --> = 3 , --> = 0 , --> = 2 , --> = 0

1. **Como caracterizaria um sistema ideal de armazenamento de ficheiro (*file hoarding*)?**

**R:** Um sistema ideal de armazenamento de arquivos (file hoarding) deve ser projetado para otimizar o uso de espaço local e garantir que os arquivos mais relevantes e necessários para o trabalho do usuário sejam armazenados de forma eficiente, mesmo em situações de desconexão ou baixa conectividade. A seguir estão as características que um sistema ideal de file hoarding deve possuir:

**Predição Inteligente e Baseada em Comportamento:** O sistema deve ser capaz de observar e aprender com o comportamento do usuário, como os arquivos frequentemente acessados e os padrões de uso. Ele deve entender que os arquivos não são acessados isoladamente, mas sim dentro de um contexto (por exemplo, projetos ou tarefas em andamento).Baseado nesse comportamento, o sistema deve prever quais arquivos serão necessários no futuro para maximizar a eficiência do armazenamento local e reduzir o desperdício de espaço.

**Distância Semântica para Determinação de Relações entre Arquivos:** O sistema ideal deve calcular a distância semântica entre arquivos para entender suas relações. Arquivos que estão logicamente relacionados (por exemplo, arquivos de um mesmo projeto ou documentos que foram acessados juntos) devem ser armazenados próximos uns dos outros, facilitando o acesso quando o usuário estiver desconectado.

**Armazenamento Baseado em Projetos:** Ao invés de armazenar arquivos individualmente, o sistema deve considerar projetos inteiros. Arquivos de um mesmo projeto ou que são frequentemente usados em conjunto devem ser armazenados juntos, o que aumenta a probabilidade de que o usuário tenha tudo o que precisa durante um período de desconexão.

**Operação sem Intervenção do Usuário:** O sistema deve ser autônomo, ou seja, sem a necessidade de intervenção constante do usuário. Ele deve ser capaz de selecionar automaticamente quais arquivos devem ser armazenados, com base no comportamento passado, sem a necessidade de o usuário fornecer listas manuais ou instruções específicas.

**Armazenamento Eficiente e Compacto:** O sistema deve ser eficiente na gestão de espaço, armazenando os arquivos mais críticos e frequentemente utilizados, enquanto evita o desperdício de espaço com arquivos raramente acessados ou desnecessários.

**Gerenciamento de Conflitos e Replicação:** Em um ambiente de replicação, o sistema ideal de file hoarding deve gerenciar eficientemente a sincronização e atualização dos arquivos entre múltiplas instâncias do sistema. Isso garante que o arquivo armazenado localmente esteja sempre atualizado em relação à versão mais recente disponível no servidor ou em outros dispositivos.

**Baixa Intervenção do Usuário:** O sistema deve ser projetado para ser o menos intrusivo possível, com pouca ou nenhuma necessidade de interação do usuário além de uma notificação de desconexão iminente. O sistema deve ser capaz de se ajustar automaticamente às mudanças nas necessidades do usuário, tornando o processo de armazenamento transparente.

Essas características tornam o sistema ideal de file hoarding altamente eficiente, inteligente e fácil de usar, adaptando-se ao comportamento do usuário e garantindo que ele tenha acesso aos arquivos necessários durante períodos de desconexão.

1. **Uma das possíveis técnicas para lidar com a perda de conexão de rede (ou conectividade fraca) é baseada na replicação e é designada por staging. Considere essa , quanto a mesma depende da correcção/exactidão (correctness) do conjunto de trabalho que é replicado? Como esta técnica lida com o cenário em que o utilizador tenta acessar um ficheiro que não está preparado? Descreva quais aspectos de segurança são considerados e como são tratados. (desenhe uma figura que mostre como o sistema funciona para explicar melhor sua resposta).**

**R:**

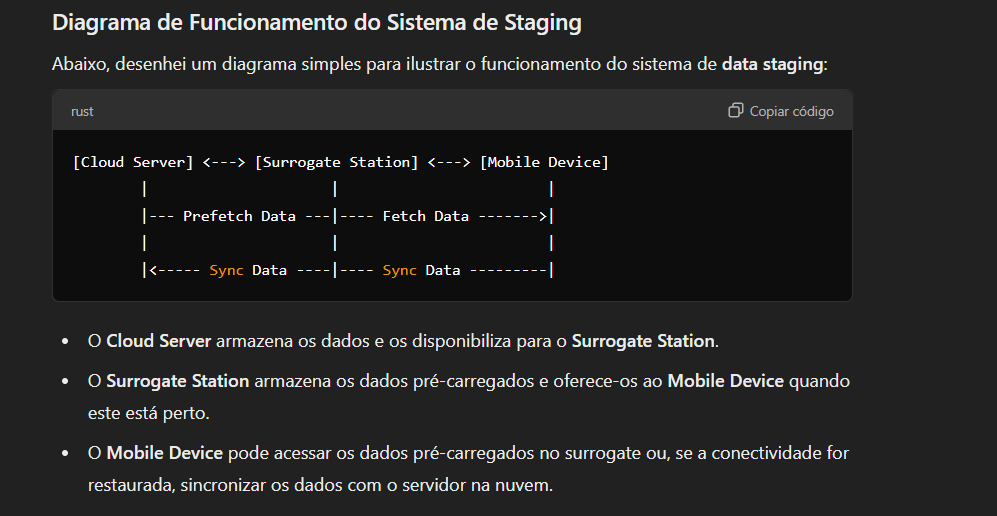
A correção do conjunto de trabalho replicado depende da precisão das previsões feitas pelo sistema sobre quais dados serão necessários, ou seja, o sistema precisa prever corretamente quais arquivos a aplicação móvel precisará acessar em um futuro próximo, para que esses dados sejam replicados para o local de staging.A exatidão do conjunto de dados replicado é crucial, pois se o sistema falhar em prever os dados necessários ou armazenar dados incorretos ou desatualizados, o usuário poderá tentar acessar arquivos que não estão presentes no surrogate ou estação de dados. Isso pode resultar em erro de acesso ou inconsistência de dados, impactando a experiência do usuário.

Se o usuário tentar acessar um arquivo que não foi prefetchado ou que não está presente no surrogate, o sistema pode adotar diferentes abordagens dependendo da implementação específica. Algumas soluções incluem:

* **Solicitação ao Servidor Remoto:** Se o arquivo não estiver no surrogate e o sistema puder estabelecer uma conexão de rede (mesmo que limitada), o arquivo pode ser recuperado diretamente da nuvem ou de outra fonte remota.
* **Fallback para Soluções Locais:** Se a conectividade for completamente perdida, o sistema pode fornecer uma experiência degradada para o usuário, como permitir acesso limitado a versões anteriores do arquivo ou alternativas de dados temporários.
* **Alertas ao Usuário:** O sistema pode notificar o usuário de que o arquivo desejado não está disponível devido a falta de conectividade ou porque não foi prefetchado, dando opções para o usuário, como aguardar a sincronização ou tentar de novo mais tarde.

Aqui estão alguns aspectos de segurança:

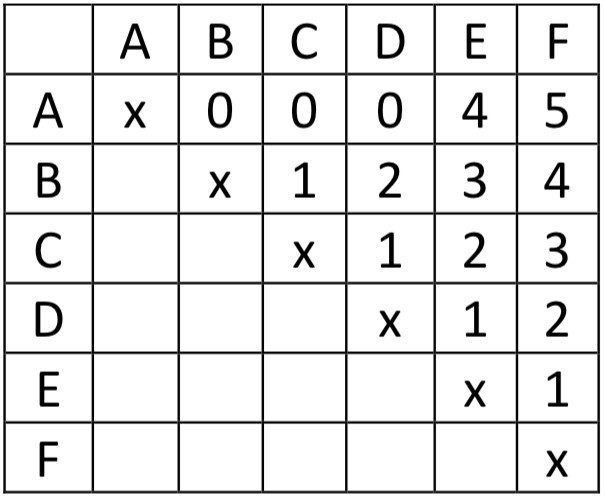
* **Autenticação e Autorização:** O sistema precisa garantir que apenas usuários autorizados possam acessar dados replicados e pré-carregados. Isso pode ser feito utilizando tokens de autenticação ou chaves de criptografia.
* **Criptografia:** Os dados armazenados localmente nos surrogates devem ser criptografados para evitar acesso não autorizado, especialmente se os surrogates estiverem em locais públicos ou não totalmente sob controle.
* **Integridade dos Dados:** O sistema deve garantir que os dados não foram alterados ou corrompidos enquanto estavam armazenados no surrogate. Isso pode ser feito utilizando somente leitura ou verificações de hash para garantir que os dados não foram modificados.
* **Sincronização de Dados:** Quando a conectividade for restaurada, o sistema precisa garantir que a sincronização dos dados entre o dispositivo móvel e o servidor de nuvem seja feita de forma segura, sem risco de overwriting ou perda de dados.´

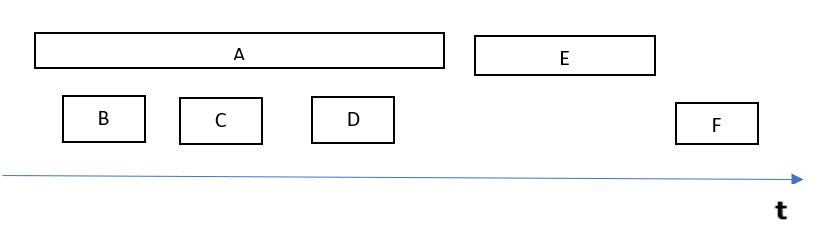


1. **Considere a técnica designada de acumulação (hoarding) que é aplicada no seguinte cenário: um utilizador leva seu laptop para um local onde a rede é inexistente, muito cara ou muito lenta, e ainda quer continuar o trabalho que estava a fazer imediamente antes de sair de seu escritório (onde seu laptop estava bem conectado à rede fixa). Nesse cenário, quão eficaz seria uma solução de *hoarding* baseada na útlima política usada recentemente?**

**R:** A técnica de hoarding baseada na política de Última Política Usada Recentemente (LRU) pode ser eficaz em cenários onde um usuário precisa continuar seu trabalho imediatamente após sair de um local com boa conectividade para um local sem rede ou com rede lenta. A LRU mantém os dados mais recentemente acessados no dispositivo, o que é útil se o usuário continuar trabalhando com os mesmos arquivos ou programas imediatamente após sair. No entanto, essa abordagem tem limitações, pois não leva em consideração dados que não foram acessados recentemente, mas que ainda podem ser essenciais para o trabalho futuro. Portanto, se o usuário precisar de informações ou arquivos específicos que não foram usados recentemente, eles podem ser removidos pela LRU, deixando o usuário sem acesso a esses dados. Uma solução mais eficaz poderia combinar hoarding com uma previsão baseada no comportamento ou nas necessidades futuras do usuário, acumulando dados importantes, mesmo que não tenham sido acessados recentemente, garantindo que os arquivos necessários estejam disponíveis mesmo sem conexão de rede.

1. **Considere a próxima tabela mostra a distância semântica de tempo de vida (*filetime semantic distance*) entre um conjunto de ficheiros. Desenhe um diagrama temporal indicando os instantes em que as operações de abrir e fechar são executadas em cada ficheiro.**



**R:** Linha de tempo das operações "abrir" e "fechar" nos ficheiros A, B, C, D, E e F. ****