

UNIVERSIDADE FEDERAL DO PARANÁ

PROJETO DE QUALIFICAÇÃO DO ECOSISTEMA DE SOLUÇÕES DIGITAIS
PARA MAPEAMENTO E GESTÃO CULTURAL

GUIA DE ATUALIZAÇÃO DE VERSÃO DO SOFTWARE MAPAS CULTURAIS

CURITIBA

2024

PROJETO DE QUALIFICAÇÃO DO ECOSISTEMA DE SOLUÇÕES DIGITAIS
PARA MAPEAMENTO E GESTÃO CULTURAL

GUIA DE ATUALIZAÇÃO DE VERSÃO DO SOFTWARE MAPAS CULTURAIS

Produto 4 da Meta 2 do Termo de Execução Descentralizada Nº 948252, firmado entre o Ministério da Cultura e a Universidade Federal do Paraná para desenvolvimento do projeto de ensino, pesquisa e extensão "Qualificação do Ecossistema de Soluções Digitais para Mapeamento e Gestão Cultural".

CURITIBA

2024

IDENTIFICAÇÃO DO PRODUTO NO TED 948252

- META 2** Publicação de uma versão atualizada do Mapas, contemplando as funcionalidades mínimas necessárias para atender à demanda por ferramentas digitais de mapeamento, monitoramento, prestação de contas, avaliação, cadastro e inscrição de propostas da Política Nacional de Fomento à Cultura Aldir Blanc.
- PRODUTO 4** 1 (um) documento sobre o processo de atualização da versão software para mais atual incluindo banco de dados

Publicação disponível no repositório público do Laboratório de Cultura Digital Brasileira (LabCDBr/gestao):

<https://github.com/LabCDBr/gestao/tree/main/produtos>

RESUMO

Este guia apresenta as boas práticas para atualização do software Mapas Culturais. O documento busca fornecer diretrizes e procedimentos básicos para facilitar a transição para novas versões da plataforma, garantindo a continuidade dos serviços e minimizando interrupções. A modernização e atualização contínua do software são cruciais para qualquer organização para garantir benefícios significativos em termos de segurança, desempenho e usabilidade.

Projeto de Qualificação do
Ecosistema de Soluções Digitais
para Mapeamento e Gestão Cultural

Guia de migração de versão do software Mapas Culturais



MINISTÉRIO DA
CULTURA





Projeto de Qualificação do
Ecosistema de Soluções Digitais
para Mapeamento e Gestão Cultural

Guia de migração de versão do software Mapas Culturais

Curitiba, 2024

Expediente

Ministério da Cultura

Secretaria Executiva

Márcio Tavares dos Santos
Secretário Executivo

Cassius Antonio da Rosa
Secretário Executivo Adjunto

Subsecretaria de Gestão Estratégica

Letícia Schwarz
Subsecretária de Gestão Estratégica

Juliana Silva Almeida
Coordenadora de Acompanhamento de Projetos

Sofia Leonor Von Mettenheim
Coordenadora-Geral de Projetos Estratégicos

Alexandre Santos
Servidor Federal

Universidade Federal do Paraná

Pró-Reitoria de Extensão e Cultura
Unidade gestora do projeto

Projeto de Qualificação do Ecossistema de Soluções Digitais para Mapeamento e Gestão Cultural

Deborah Rebello Lima
Coordenadora-Geral

Caio Eduardo Ferreira de Miranda
Bolsista - Desenvolvimento

Kauane Gonçalves de Souza Batista
Bolsista - Desenvolvimento

Lúcia Helena Alencastro
Vice-Coordenadora

Danilo Amorim Rabêlo
UX/UI Designer

Leandro Franklin Gorsdorf
Bolsista Docente (Pesquisa/Extensão)

Uiraporã Maia Carmo
Coordenador Técnico

Diego Figueiredo Rojas
Designer

Leonardo José Zanotti
Bolsista - Desenvolvimento

Lucas Pirola Dias
Líder de Tecnologias

Elliott Victor de Sousa Chaves
Cientista de Dados

Lia Vieira Ramalho Bastos
Bolsista Técnico

Ranielder Fábio de Freitas
Líder de Design

Emille Catarine Rodrigues Cançado
UX/UI Designer

Luiz Augusto Fernandes Rodrigues
Bolsista Docente (Pesquisa/Extensão)

Rodrigo Gomes Marques Silvestre
Líder de Governança e Narrativas

Enzo Spironelli Pinheiro
Bolsista - Desenvolvimento

Marília Teixeira Gomes
Bolsista Técnico

Washington Forte da Silva
Gestor de Projeto

Fábio Naconeczny da Silva
Bolsista - Desenvolvimento

Márlcio Esmeraldo Ribeiro
Analista de Comunicação

Aiumy Machado Alberti
Bolsista - Comunicação e Design

Gabriela Araceli Britez Timm
Bolsista - Mapeamento e Gestão

Mayala Tereza Fernandes
Bolsista - Comunicação e Design

Alexandre Ribeiro Cajazeira Ramos
Cientista de Dados

Gabriela Morais Gandine
Bolsista - Desenvolvimento

Nara Cristina Moura Pessoa
Analista de Comunicação

Alyssa Riccieri Oliveira Santos
Bolsista - Mapeamento e Gestão

Heitor Plinta de Oliveira
Bolsista - Desenvolvimento

Patrick de Oliveira Lemes
Bolsista - Desenvolvimento

Ana Beatriz Polena Prendin
Bolsista - Mapeamento e Gestão

Jonathan Fabricio Delgado
Bolsista Técnico

Roberto Martins de Jesus
Bolsista Técnico

Angela Roberta Lucas Leite
Bolsista - Governança

Juliana Mara Lima das Neves
Bolsista Técnico

Vanuza Aparecida Santos Wistuba
Analista de Comunicação

Anne Caroline T. da Silva Strapaçon
Bolsista - Gestão de Projetos

Juliane Brito Scoton de Souza
Designer de Serviço

Victor Magalhães Pacheco
Programador Backend



Sumário

Apresentação.....	5
Contexto, história e processo de desenvolvimento do software.....	6
Versões do software e modificações no banco de dados.....	7
Avaliação da necessidade de atualização.....	7
Estratégia de atualização.....	9
Código.....	9
Banco de dados.....	10
Ambientes: testes, homologação e produção.....	10
Riscos e plano de contingência.....	12
Perfil de equipe.....	12
Testes e validação.....	13
Preparação.....	14
Definição de escopo.....	14
Execução.....	14
Validação de dados.....	14
Avaliação de desempenho.....	15
Métricas de desempenho.....	15
Implementação.....	15
Comunicação.....	16
Suporte e documentação.....	17
Evolução contínua.....	18



Apresentação

Este documento tem como objetivo servir de base inicial para construção colaborativa de um guia de orientações para a atualização de soluções digitais de fomento à cultura, iniciando especificamente pelo software livre Mapas Culturais, utilizado no Sistema Nacional de Informações e Indicadores Culturais. Com diretrizes claras e procedimentos básicos, este guia visa facilitar o processo de transição para novas versões da plataforma, garantindo a eficiência e minimizando a interrupção dos serviços.

Com a reestabilização do Ministério da Cultura, o principal software de gestão cultural do país também está sendo revigorado. Nesse contexto, ações iniciais estão sendo implementadas para melhorar a eficiência e eficácia do software, a fim de atender melhor às necessidades da gestão cultural.

Modernizar e manter o software atualizado traz múltiplos benefícios para qualquer organização. No caso do Ministério da Cultura, a atualização do software irá melhorar significativamente a segurança, a performance e a interface do sistema. Estas melhorias irão não só aumentar a eficiência e a produtividade, mas também proporcionarão uma experiência de usuário mais agradável e intuitiva para aqueles que interagem com o software nos servidores do MinC.





Contexto, história e processo de desenvolvimento do software

O desenvolvimento do software ao longo dos mais de 10 anos de existência passou por várias fases e momentos como maior ou menor colaboração, maior ou menor desenvolvimento de novas funcionalidades, etc.

Os ciclos também ficam evidentes no código, é possível visualizar códigos melhores ou piores documentados, mais ou menos organizados, etc. O que significa que existem partes que estão mais sólidas do que outras, mais fáceis de manter do que outras, e que por sua vez possuem menos problemas que outras.

O desenvolvimento teve diversos financiadores ao longo do tempo e todos eles elevaram o software a um novo patamar, e fez com que mais pessoas se conectassem com o ecossistema do software.

O software atual é escrito principalmente em PHP, incluindo html, css e javascript. Além das linguagens de programação algumas bibliotecas são importantes destacar, como o Slim Framework no backend e o VueJS no frontend. Além claro do banco de dados PostGIS. O grande desafio na atualização é garantir que as modificações customizadas pelos temas e plugins estejam compatíveis com as novas versões ou então será necessário descontinuar a utilização dos mesmos.

Versões do software e modificações no banco de dados

Entre a versão 5 e 7 do mapas mais de 5000 commits foram feitos, mais de 3000 arquivos alterados, resultando em muitas novas funcionalidades e correções.

As principais funcionalidades implementadas incluem um botão de aplicar avaliações para avaliações técnicas, exibição de parecer na avaliação documental e técnica, um campo de dados bancários vinculado ao agente responsável e um botão que sincroniza as inscrições entre as fases. As melhorias incluem a instalação de uma nova biblioteca para geração de arquivos CSV, novos componentes Vue.js para tabelas, refatoração da listagem de inscrições e a implementação de melhorias de desempenho.

São numerosas as correções de erros, tais como correção de seleção de relacionamentos OneToOne, correção de erros de instalação do npm após o node, correções de extração de planilhas em Excel das entidades via API, ajustes na ordem alfabética dos tipos de entidades e correções na configuração do corte da imagem do avatar.

As bibliotecas do php e do javascript também foram atualizadas para novas versões, além de uma atualização para suporte da versão mais atual do php a 8.

Avaliação da necessidade de atualização

A atualização de software é uma tarefa significativa que requer uma avaliação prévia cuidadosa. Antes de tomar a decisão de migrar, é preciso analisar a necessidade real e identificar os desafios, os benefícios, assim como os possíveis contratempos de tal tarefa. O





sistema de software atual atende suficientemente às necessidades do negócio? Existem problemas críticos que apenas um novo sistema pode resolver? Estas são algumas das questões que devem ser respondidas durante a avaliação.

No contexto atual, é primordial que os tomadores de decisão e gestores estejam bem informados sobre as implicações da atualização. Isso envolve a educação e o diálogo sobre os prós e os contras, assim como as possíveis repercussões. Os gestores também devem estar cientes do tempo, custos, recursos e outros aspectos ligados ao processo.

A compreensão clara das vantagens e desvantagens, é essencial. Que benefícios traz essa atualização para o contexto em questão? Serão novas funcionalidades melhoradas, maior eficiência, redução de custos ou mais segurança? Por outro lado, o que possivelmente será perdido? São dados ou funcionalidades específicas, ou talvez a familiaridade com o sistema atual? Ao avaliar esses fatores, será possível fazer uma escolha informada balançando os benefícios e os contrapesos.

Qualquer processo de atualização de software trará consigo desafios. A complexidade do processo será geralmente dependente do tipo de software, dos dados envolvidos e da escala. Alguns dos desafios a serem enfrentados podem incluir a transferência segura e eficiente de dados, a minimização do tempo de inatividade, a formação de funcionários no uso do novo software, entre outros.

Embora seja importante ponderar quando efetuar atualizações de software, é essencial manter-se atualizado. As atualizações de software não só proporcionam novas funcionalidades e melhorias, mas também corrigem problemas e resolvem questões de segurança. Por esse motivo, é recomendado progredir constantemente, adaptar-se às mudanças e evoluir de acordo com a evolução do software e as necessidades da organização. Ponderando cada atualização, que deve ser avaliada cuidadosamente para

garantir que beneficia realmente a organização e não causará repercussões indesejadas.

Estratégia de atualização

Deverá ser elaborado um plano estruturado para realizar a atualização, que engloba detalhes técnicos cruciais relativos à implementação da atualização. Este plano fornecerá um desdobramento faseado do processo, incluindo uma linha do tempo com marcos significativos para assegurar a conclusão da atualização. Além disso, serão identificados quais recursos, tanto humanos quanto tecnológicos, serão essenciais durante essa transição.

Importante ressaltar, a atualização também envolverá considerações sobre a compatibilidade entre o software e a base de dados. Este componente vital do plano garantirá que tanto o software quanto o banco de dados possam funcionar em conjunto sem problemas, evitando conflitos de integração que possam impactar o desempenho geral do sistema. Dessa forma, atenderemos com eficiência às necessidades do sistema e garantimos sua operação sem interrupções.

Código

Uma cópia do software Mapas Culturais pode ser feita no endereço <https://github.com/culturagovbr/mapasculturais>, essa etapa consiste em baixar o código-fonte no servidor onde a instalação deve acontecer, deverá ser realizada a preparação do servidor para receber o software, assim como a configuração.

Uma avaliação abrangente das alterações no código-fonte é necessária ser feita entre a versão que está atualmente, em muitos





casos é a versão 5 e a versão mais utilizada, no momento de escrita desse documento é a versão 7.4. A natureza dessas alterações têm impacto direto na atualização do software e é importante fazer uma avaliação das principais funcionalidades utilizadas e o que mudou entre uma versão e outra, a fim de garantir que as modificações presentes nas funcionalidades sejam de conhecimento de todos da equipe, tanto de pessoas desenvolvedoras quanto gestoras e agentes culturais.

Banco de dados

A abordagem básica envolve tentar atualizar diretamente do banco de dados da versão 5 para a versão 7. No entanto, cabe destacar que se houver muitas modificações presentes, isso pode influenciar o custo da migração, tornando-a mais significativa. Em alguns casos, pode ser necessário abandonar algumas dessas modificações para facilitar a atualização.

Para minimizar a interrupção dos serviços durante a atualização, adotaremos uma série de procedimentos pré-planejados. Isso inclui executar a atualização durante períodos de baixa atividade do usuário, manter um canal aberto de comunicação com os usuários sobre o status da atualização, e estar preparado para implementar rapidamente a versão anterior do software, se necessário. Esses procedimentos fornecem um escudo contra o tempo de inatividade excessivo e interrupções de serviço indesejadas, preservando a experiência ideal do usuário durante todo o processo de atualização.

Ambientes: testes, homologação e produção

Esses são os passos para atualizar o código fonte de um software em diferentes ambientes, como testes, homologação e produção. Os dados de cada ambiente devem incluir o banco de dados completo

com dados reais. Você pode encontrar os comandos para atualizar o banco de dados no repositório do projeto do software.

1. Fazer backup do banco de dados é prioritário, atualizar o código fonte através de um release que pode ser encontrado no link https://docs.mapasculturais.org/mc_deploy_update/. Confira a versão mais recente e compare com a versão que você está usando. É importante visitar a pasta de documentação "Release notes" para verificar as anotações das versões entre a atual e a mais recente. Pode haver procedimentos específicos ou precauções a serem tomadas.
2. Uma vez preparado, é possível começar a atualizar o código. Usando o comando 'git fetch', você pode atualizar as referências do novo código. Para alterar para uma nova versão estável, utilize 'git checkout [release]', onde '[release]' é o release desejado.
3. O próximo passo é executar o script de atualização 'deploy.sh' localizado na pasta de scripts. Este script atualiza as bibliotecas e dependências da aplicação e também atualiza o banco de dados com novos campos ou tabelas. Após este passo, é necessário reiniciar o serviço PHP para que todos os arquivos sejam reescritos corretamente. Isso requer permissão de root.
4. No caso de problemas, como problemas de permissão ou dificuldades com 'git pull', existem soluções disponíveis. Certifique-se de que todos os comandos sejam executados com o mesmo usuário da aplicação do software. Se você encontrar qualquer problema diferente, considere entrar em contato com a comunidade do software para obter assistência.





Riscos e plano de contingência

Em geral, processos de atualização possuem riscos, alguns são previsíveis, outros inesperados. É crucial ter um plano de contingência robusto e minuciosamente preparado para orientar a transição dos dados para a nova versão do software. Este plano deve detalhar cada etapa, garantindo que a integridade dos dados seja preservada em todos os momentos. Isso envolve a realização de backups completos de todos os dados antes do processo de atualização. Em caso de perda de dados, devem ser implementadas estratégias confiáveis de recuperação de dados que permitirão a restauração rápida e precisa das informações. Este plano de contingência busca garantir maior suavidade e sem interrupções, minimizando os riscos e garantindo a continuidade dos serviços.

Uma medida essencial para garantir a integridade dos dados e ter uma maior suavidade no processo de atualização é seguir a documentação de cada versão "major", versão por versão. Por exemplo, se a versão atual é a 7 e a instalação a ser atualizada ainda está na 4, primeiro a atualização deve ser feita para versão 5, depois para 6 e só então para 7. Cada versão "major" possui modificações que precisam que aconteçam antes de atualizar para o próxima. Com todo processo delicado e sensível, diminuir ele em etapas menores, garante que os passos dados são sólidos e será possível seguir com maior tranquilidade. Essa é só uma das formas de diminuir os riscos nos processos de atualização e o plano de contingência deve englobar outros para diminuir o máximo possível os erros inesperados no processo.

Perfil de equipe

Para fazer a atualização de software que implica a migração de um banco de dados e a reorganização dos arquivos salvos pela aplicação, você precisará de um time de especialistas que possua



um mix de habilidades para lidar com todas as necessidades que possam aparecer. Abaixo três perfis recomendados para uma transição mais tranquila.

- 1. Desenvolvedor Web especialista na linguagem PHP:** Essa pessoa seria responsável pelo desenvolvimento e manutenção do backend do seu software, utilizando PHP e o framework Slim. Eles possuem conhecimento sólido em PHP e conexão com banco de dados PostGIS, o que será benéfico quando estiver migrando dados da antiga estrutura para a nova.
- 2. Administrador de Banco de Dados (DBA) especializado em PostGIS:** Esse profissional tem experiência específica em trabalhar com os dados brutos e será encarregado de gerenciar o banco de dados durante a atualização do software. Eles saberão como migrar os dados com segurança e eficácia, garantindo que nada seja perdido ou danificado no processo.
- 3. Gerente de Projeto:** Esta pessoa é responsável por gerenciar a equipe, planejar, organizar e controlar recursos para alcançar os objetivos do projeto. Eles também teriam habilidades de resolução de problemas abrangentes e uma sólida compreensão da infraestrutura de TI para garantir que a migração seja suave e que a nova estrutura esteja funcionando corretamente após a atualização.

Testes e validação

A metodologia usada para testar o software atualizado e validar a integridade dos dados existe e pode ser aproveitada para iniciar a realização dos testes. A seguir, um panorama das principais objetivos dessa etapa:



Preparação

- **Definição de escopo:** identificar todos os componentes do sistema a serem testados, incluindo novas funcionalidades e alterações na lógica existente.
- **Definição de ambientes de teste:** estabelecer ambientes de teste que simulem o ambiente de produção o mais próximo possível.

Definição de escopo

- **Casos de teste funcionais:** criação de casos de teste detalhados que cobrem as funcionalidades do sistema, tanto novas quanto existentes.
- **Casos de teste de integração:** verificação da correta interação entre os diferentes módulos e componentes do sistema.

Execução

- **Executar testes de integração:** validar que os componentes do sistema trabalham corretamente em conjunto.
- **Executar testes de sistema:** testar o sistema como um todo, incluindo interações com banco de dados, APIs externas, etc.

Validação de dados

- **Comparação de dados antes e depois:** comparar snapshots do banco de dados antes e depois da atualização para detectar inconsistências.
- **Verificação de integridade:** utilização de scripts SQL ou ferramentas de terceiros para verificar a integridade dos dados.



Avaliação de desempenho

- **Testes de carga:** simular múltiplos usuários acessando o sistema simultaneamente para identificar gargalos de desempenho.
- **Testes de stress:** testar o sistema até seus limites para ver como ele se comporta sob condições extremas.

Métricas de desempenho

- **Tempo de resposta:** medir o tempo de resposta para várias transações e operações do sistema.
- **Uso de recursos:** monitoramento de CPU, memória e disco durante os testes de carga e estresse.
- **Número de erros e exceções:** registro do número de erros/exceções encontradas durante os testes.

Implementação

Uma vez o roteiro de testes validado, você planeja a atualização desde a data que irá ocorrer, preparar equipe de suporte, atualização documentação, verificar e agendar o tempo de indisponibilidade da plataforma, verificar processos de inscrição abertos e com alguma utilização no período agendado para execução da atualização.

No dia D, é importante ter equipe de plantão, não só a técnica mas de negócio para serem os primeiros a testar depois de finalizar a implementação a fim de minimizar o impacto nos usuários.



Comunicação

O processo de atualização de software deve começar com a comunicação clara para os usuários dos planos de atualização iminente. Isso inclui enviar avisos de atualização para permitir que os diversos perfis de usuários se planejem adequadamente. Esses avisos devem incluir quando a atualização está programada para começar, sua duração estimada, bem como qualquer impacto potencial que possa ter sobre a operação normal do software.

Para manter os usuários informados sobre o progresso da atualização, devem ser enviadas comunicações regularmente, incluindo quando a atualização começar, acabar, e cada vez que uma nova atualização ocorrer. Isso ajuda a garantir que os usuários estejam sempre atualizados sobre o status do software e possam planejar suas atividades diárias de acordo.

Além disso, é importante elaborar conteúdo informativo sobre as melhorias feitas em cada atualização do software e quaisquer problemas que tenham sido corrigidos. Esses relatórios, geralmente chamados de "release notes", fornecem aos usuários uma visão detalhada das alterações feitas e como essas alterações irão beneficiá-los. Ele também aumenta a transparência e ajuda a construir a confiança entre a equipe de desenvolvimento do software e os usuários.

A instituição também deve considerar a utilização de todos os seus canais de comunicação disponíveis para divulgar a realização da atualização. Isso poderá incluir o uso de mídias sociais, e-mails, newsletters e outros canais de comunicação interna adequados.

É crucial que os usuários sejam informados sobre as novas funcionalidades e quaisquer alterações na interface do usuário por meio de sessões de treinamento e documentos de orientação atualizados. Eles podem precisar de ajuda para se adaptarem à nova

versão do software e compreenderem completamente como usar todas as novas funcionalidades.

Para auxiliar com isso, recursos adicionais devem ser fornecidos, tais como manuais de usuários atualizados, guias "how-to", vídeos de treinamento e outros materiais de aprendizagem relevantes.

Finalmente, a instituição pode considerar a realização de sessões de perguntas e respostas com os usuários para abordar quaisquer dúvidas persistentes ou problemas que possam surgir ao usar a nova versão do software. Essas sessões podem ser vitais para garantir a adoção suave do software atualizado pelos usuários e para fornecer feedback valioso à equipe de desenvolvimento de software para melhoramentos futuros.

Suporte e documentação

A primeira etapa importante envolve a atualização da documentação e dos materiais de orientação para os usuários. É crucial que todas as informações sejam atualizadas para refletir as mudanças feitas no software. Isso não só ajuda os usuários a se adaptarem ao novo sistema, mas também garante que eles tenham acesso a informações corretas e atualizadas. Isso pode incluir a revisão de manuais de usuários, a atualização de guias de procedimentos e a revisão de quaisquer outros materiais de referência que os usuários possam precisar ao usar o software.

O próximo passo seria a criação de um FAQ para responder às principais questões relacionadas à atualização. Este recurso pode ser extremamente valioso para os usuários, pois eles podem ter muitas perguntas sobre o que esperar durante e após o processo de atualização. Um FAQ detalhado pode abordar questões como o tempo esperado de inatividade, como os dados serão afetados, o que





os usuários precisam fazer para se preparar, entre outras possíveis questões.

Outra tarefa importante seria o estabelecimento de um canal exclusivo para consulta e suporte durante e após o processo de atualização. Este canal forneceria aos usuários um meio direto de comunicação com a equipe de suporte para tirar dúvidas e relatar problemas. Ter um canal específico para o processo de atualização ajudaria a garantir que as questões são tratadas de maneira eficiente e eficaz, sem se misturar com outras solicitações de suporte.

Finalmente, o estabelecimento de uma agenda de suporte e manutenção pós-atualização é imprescindível. Esta agenda deve incluir detalhes sobre como serão tratadas as solicitações de suporte, como os problemas após a atualização serão gerenciados, e o plano para futuras atualizações do software. Esta comunicação clara e detalhada com os usuários não apenas estabelece as expectativas corretamente, mas também fornece um plano claro sobre o que eles podem esperar em termos de suporte e atualizações futuras. Uma abordagem transparente e estruturada neste ponto também ajudará a construir a confiança dos usuários no novo software.

Evolução contínua

A manutenção preventiva e de segurança em todos os softwares do ambiente é um passo crucial e contínuo. Manter todos os sistemas atualizados minimiza o risco de problemas e falhas de segurança. Isso envolve regularmente a inspeção, atualização e reparo de todos os componentes de software. Além disso, monitorar os sistemas em busca de sinais de possíveis falhas e corrigir problemas antes que eles se tornem críticos também é uma parte crucial desta prática. Ao fazer isso, você garante que todos os aplicativos, sistemas operacionais e outros softwares em sua infraestrutura estão operando de maneira ideal e segura.



Em relação à política de backups, uma estratégia sólida é indispensável para qualquer organização. Não se trata apenas de fazer backups regulares, mas de assegurar que os dados sejam adequadamente protegidos e facilmente recuperáveis em caso de perda de dados. Isso envolve a escolha do tipo correto de backup (como backup total, incremental ou diferencial), a identificação dos dados que precisam ser protegidos, o estabelecimento de um cronograma de backup, a implementação de protocolos de segurança para proteger os backups e a realização de testes regulares para garantir que os dados possam ser recuperados com sucesso quando necessário. Garantir que a Lei Geral de Proteção de Dados (LGPD) esteja contemplada na política de backup e arquivamento de dados.

Manter-se atualizado com as atualizações e versões mais recentes da comunidade é essencial. Acompanhar as últimas notícias, recursos e mudanças de plataforma pode ajudá-lo a aproveitar ao máximo seu software e evitar problemas potenciais. A comunidade é um grande recurso para aprender melhores práticas, obter conselhos e suporte técnico, e manter-se atualizado sobre as últimas tendências e atualizações de software.

A prática de "Release Early, Release Often" implica em compartilhar regularmente as melhorias do software com a comunidade. Ao introduzir novas versões frequentes e menores no software, os usuários e a comunidade podem contribuir com o desenvolvimento com feedback e sugestões valiosas. Isso também cria uma sensação de progresso contínuo e transparência que gera confiança e lealdade entre os usuários e a comunidade. Além disso, ao garantir que todas as melhorias do software estejam de acordo com a licença AGPLv3, promove-se a livre distribuição e utilização do software criado.

Projeto de Qualificação do Ecossistema de
Soluções Digitais para Mapeamento e Gestão Cultural

REALIZAÇÃO



MINISTÉRIO DA
CULTURA



APOIO

