 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Sumário

1.	OBJETIVO.....	1
2.	ÂMBITO DE APLICAÇÃO	1
3.	DEFINIÇÕES.....	Erro! Indicador não definido.
4.	DOCUMENTOS DE REFERÊNCIA.....	1
5.	RESPONSABILIDADES	2
6.	REGRAS BÁSICAS	2
6.	CONTROLE DE REGISTROS.....	14
7.	ANEXOS.....	15
8.	REGISTRO DE ALTERAÇÕES	15

1. OBJETIVO

O Objetivo desta GED se dá pelas definições que o Grupo CPFL adotará para ditar as boas práticas de Arquitetura e Desenvolvimento de Softwares.

2. ÂMBITO DE APLICAÇÃO

2.1. Empresa

Será aplicado para todo âmbito do Grupo CPFL sempre que houver algo relacionado a demandas corretivas, demandas de melhoria e projetos de tecnologia da informação.

2.2. Área


Aplicado a todas as áreas corporativas e que desenvolvem atividade administrativa para o Grupo CPFL.

3. DEFINIÇÕES

3.1. CÓDIGO LEGADO

Fica convencionado que aplicações e sistemas que são mantidos para efeitos de histórico ou que não tem atualização tecnológica nos últimos **cinco anos** serão considerados parte do parque de sistemas **LEGADOS** do Grupo CPFL.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	1 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

A manutenção do código **LEGADO** deverá ter uma análise profunda e criteriosa devido a possibilidade de não haver documentação necessária para embasamento, sendo assim um **risco** para as dependências de negócio atreladas ao mesmo.

4. DOCUMENTOS DE REFERÊNCIA

[OWASP SAMM | OWASP Foundation](#)

[OWASP Proactive Controls | OWASP Foundation](#)

Norma de Desenvolvimento Seguro - GED 18883

Diretrizes de Segurança da Informação - GED 14369

Norma Básica de Segurança da Informação - GED 18704

5. RESPONSABILIDADES

É de responsabilidade da equipe de Arquitetura e Dados a aplicabilidade das definições contidas nesse documento e que o mesmo deverá ser aplicado para todo escopo de tecnologia da informação a partir da publicação.

6. REGRAS BÁSICAS

As definições deverão pautar as discussões sobre organização, sustentação do Grupo CPFL assim como amparar futuros projetos.

6.1. MANUTENÇÃO DE CÓDIGO LEGADO

6.1.1. CUIDADOS


O Grupo CPFL possui diversas aplicações/sistemas que se identificam como legado, conforme explicação do item [3.1.](#). Esses sistemas possuem pouca documentação ou baixa taxa de atualização de documentação após inúmeros anos de demandas e projetos. Sendo assim deverá haver grande dedicação e cuidado com qualquer alteração de código dos mesmos.

Os cuidados deverão ser tomados com base no seguinte:

6.1.1.1. ATUALIZAÇÃO DE DOCUMENTAÇÃO

A Atualização de Documentação de sistemas LEGADOS deverá ser feita integrando o conteúdo novo a ser implantado que será incrementado as regras já existentes.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	2 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

6.1.1.1.1. CRIAÇÃO DE DOCUMENTAÇÃO

Havendo a eventualidade de que os sistemas LEGADOS não tenham documentação anterior a demanda em questão, deverá ser também escopo da demanda.

É imprescindível que haja aprovação da documentação por parte da área de negócio, para que não haja divergência entre o código e as reais necessidades atendidas pelo sistema.

6.1.1.2. AVALIAÇÃO DE RISCOS

Sempre que houver demanda ou projeto que contenha alteração, melhoria e evolução de sistemas e aplicações que estejam configuradas como LEGADO será necessário que junto as documentações a serem entregues a sustentação da CPFL também seja entregue uma avaliação dos riscos.

A Documentação deverá conter os impactos e riscos dos seguintes itens a seguir:

- Infraestrutura Existente
- Regras de Negócio

6.1.1.3. MAPEAMENTO DE PROCESSOS ASSOCIADOS

Deverá ser feito também um estudo de processos associados ao projeto ou demanda que está sendo desenvolvido no escopo LEGADO.

Esse estudo estará fortemente associado ao item [6.1.1.2](#), e gerará uma visão de continuidade as regras já existentes.

6.1.2. MANUTENÇÃO DE ESTRUTURAS


Os sistemas do Grupo CPFL possuem estruturas já consolidadas e que oferecem um grande risco se forem alteradas.

6.1.2.1. ESTRUTURA

Alguns sistemas da CPFL possuem sua estrutura principal com as seguintes tecnologias:

- MVC
- WebForms
- MVC/DDD
- Outras

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	3 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Alterar essas estruturas podem colocar em risco a hospedagem, complexidade de manutenção e viabilidade do código empregado. Sendo assim, só poderão ser alteradas se houver um refatoramento completo da solução ou utilização de código externo a solução.

Códigos externos a solução poderão ser utilizados no formato de micro serviços, WebServices, APIs REST.

6.1.2.2. FUNÇÕES

Sempre que houver uma função dentro do código em que a mesma é utilizada em mais de um único ponto da solução deverá ser avaliado a possibilidade de se criar um OVERLOAD da mesma. A exceção do caso se dará quando a demanda estiver com o indicativo de alteração da função na sua totalidade.

6.2. CRIAÇÃO DE NOVOS SISTEMAS

O Grupo CPFL busca sempre a informatização e a melhoria de seus processos, o que invariavelmente faz com que novos sistemas surjam no portfólio da companhia.

Esses sistemas poderão ser construídos unicamente para a CPFL ou adquirido pela mesma de fornecedores que ofertam esse tipo de produto ao mercado.

6.2.1. DESENHO DE ARQUITETURA

Todos os novos sistemas que o Grupo CPFL adquirir deverão obrigatoriamente conter os documentos a seguir para que exista um controle do Desenho da Arquitetura que foi empregado na solução.

6.2.1.1. INFRAESTRUTURA


Como melhores práticas o Grupo CPFL solicita que sempre que houver alteração em escopo de infraestrutura deverá ser feita uma entrega de documentação do desenho após implantação.

Esse desenho deverá ser incremental em caso de Demandas e Projetos de melhorias em sistemas já entregues a sustentação da TI CPFL, mantendo assim a documentação atualizada e economizando tempo de estudo para futuras iniciativas.

Projetos que criarão ou que farão a implantação de novos sistemas e plataformas deverão ter no escopo de entrega a documentação referente a infraestrutura também.

Demandas de melhoria oriundas do escopo de sustentação deverão se atentar para o [item 6.1.1.1.](#)

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	4 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Projetos e Demandas em que o produto final seja baseado em uma contratação SaaS (Software as a Service) deverão possuir em sua documentação apenas a visão macro da infraestrutura, uma vez que as responsabilidades sobre as mesmas recaem sobre os fornecedores envolvidos nos contratos.

A seguir a listagem de documentação mínima que as soluções devem conter, podendo ser incrementadas conforme necessidade de cada situação:

- Desenho da Arquitetura
- Estruturação de esteira de Infraestrutura como Código
- Estratégias de Backup
- Dicionário de Serviços (Para soluções feitas em Micro serviços)
- Dicionário de Dados
- MER dos Bancos de Dados

6.2.1.2. SEGURANÇA DA INFORMAÇÃO

Como padrão o Grupo CPFL aplicou políticas duras de segurança para que o ambiente da companhia possa não sofrer com as inúmeras tentativas de invasão e extorsão que existem hoje no mundo.

Dessa maneira foi criado uma GED pela equipe de segurança e que deverá ser seguida para que o bom andamento dos trabalhos no ambiente CPFL.

Referência:

- Norma de Desenvolvimento Seguro - GED 18883
- Diretrizes de Segurança da Informação - GED 14369
- Norma Básica de Segurança da Informação - GED 18704

6.2.1.3. INTEGRAÇÕES DE NEGÓCIO


Devido ao tamanho do escopo regulatório e de negócio em que as aplicações do Grupo CPFL atuam, se faz necessário que as documentações tragam também as integrações entre os sistemas.

Essa documentação deve vir em um arquivo de fluxo no Bizagi (ou outra ferramenta padrão que a CPFL vir a usar), onde o desenho deve ilustrar as integrações e comentar as regras e necessidades vinculadas.

6.2.2. PADRÕES DE CÓDIGO

6.2.2.1. NOMENCLATURA DE VARIÁVEIS, FUNÇÕES E NOMES DE COMPONENTES

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	5 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Para desenvolvimento de código novo a partir da aplicação deste documento, deverá ser respeitado os seguintes itens a respeito da nomenclatura de variáveis e funções:

- Não deverá ser utilizado outro idioma a não ser Português-Brasil (PT-BR) ao nomear variáveis e funções.
- Não deverá ser utilizado acentuação para nomear variáveis e funções.
- Não deverá ser utilizado espaço ao nomear variáveis e funções.
- A extensão dos nomes deverá ser limitada a um máximo de 25 caracteres.
- A nomenclatura poderá utilizar de Camel Case ('AaA').
- Não deverá ser utilizado 'Underline' ou 'Underscore' para nomenclaturas de variável ou funções.
- A nomenclatura da variável ou função devem estar associadas a sua finalidade, sendo compreensiva e racional para melhor leitura do código fonte.
 - Exemplo: Para uma função de cadastro de usuários a função poderia ter o nome de CadUsuario ou CadastroUsuario.

6.2.2.2. COMENTÁRIOS

Comentários de código deverão seguir as boas práticas a seguir:

- Os comentários não devem duplicar o código.
- Bons comentários não justificam códigos pouco claros.
- Se você não conseguir escrever um comentário claro, pode haver um problema com o código.
- Os comentários devem dissipar as confusões, não causá-las.
- Forneça links para a fonte original do código copiado.
- Inclua links para referências externas onde elas serão mais úteis.
- Adicione comentários ao corrigir bugs.
- Use comentários para marcar implementações incompletas.


Comentários utilizados nesse formato poderão ajudar a melhorar a leitura do código assim como auxiliar em atividades de documentação das soluções.

6.2.2.3. TRATAMENTO DE EXCEÇÕES

Exceções deverão ser mecanismos utilizados de maneira padronizada dentro dos códigos desenvolvidos para o Grupo CPFL.

Fica terminante proibido que uma exceção não tenha tratamento do código, ou seja, que a estrutura Try/Catch fique em branco, mascarando problemas.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	6 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Também fica proibido a gravação de dados sensíveis de usuários em Logs da aplicação, se os mesmos tiverem que ser armazenados deverão estar anonimizados.

Todo e qualquer exceção deverá gravar log em banco de dados da solução ou arquivo de texto de maneira estruturada.

6.2.2.3.1. LOGs

Os Logs das aplicações deverão conter a seguinte estrutura mínima para análise:

- Data e Hora da ocorrência, com registro até os mili segundos
- Fluxo que está gerando a exceção
- Classe com problema
- Linha de código
- Exceção com detalhe interno

Para os Logs registrados em arquivo de texto, segue deliberações:

- Os arquivos devem ter tamanho máximo restrito a 10mb
- Caso o arquivo tenha que crescer mais, deverá ser gerado um novo arquivo e o anterior arquivado de maneira organizada e sequencial.
- O nome do arquivo deverá conter o Dia em sua estrutura.
- Logs de Aplicação deverão ter nomes diferentes, acompanhando a classe de origem

6.2.2.4. ARQUITETURAS – DESIGN


6.2.2.4.1. DDD

O Grupo CPFL entende que o modelo de design baseado em Domínios (Domain Design Driven) corresponde a estruturas de código onde o ponto de decisão para criação de serviços e jornadas está atrelado ao domínio de negócio.

Dito isso, também é convencionado que não exista um design único por definição, mas diversos que podem se enquadrar baseado na necessidade do negócio e qual teria maior consonância com as diretrizes da CPFL.

Deverá ser tomado como **BASE** para análise a estrutura de Design mais comum: Presentation, Application, Domain e Infra.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	7 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Não está vetado o incremento ou decréscimo dessa estrutura, entretanto, deverá ser tratado e discutido a melhor aplicabilidade desse tipo de Design de estruturas projeto a projeto.

Essa discussão deverá gerar uma aprovação formal e que deverá constar na documentação da demanda/projeto.

6.2.2.4.2. MVC

A Estrutura Model-View-Controller divide em três camadas majoritárias e que tem a seguintes responsabilidades:

Model: Responsável por representar as informações de negócio, ou seja, acesso e manipulação de dados da aplicação.

View: Responsável pela interface que o sistema usará, mostrando as informações oriundas da Model na interface do usuário. Não é indicado que seja feita a validação de dados nesta camada.

Controller: Responsável por ligar as camadas **Model** e **View**, fazendo os dados serem entregues a camada de visualização e vice e versa.

Dentro dessa estrutura de design e arquitetura, é indicado que haja uma boa análise no momento de se utilizar funções/rotinas, para que não haja diversos casos de duplicação de código entre **Controllers** ou que uma **Controller** fique com muitas linhas de código. Se houver essa necessidade de construção de grandes e complexos trechos de código a indicação é que se faça a construção associada a uma classe própria para a atividade, centralizando a lógica de negócio e diminuindo a complexidade do código.

6.3. REPOSITÓRIO DE CÓDIGO


O Grupo CPFL possui atualmente diversos repositórios de código fonte, que envolvem suas aplicações de código LEGADO ou novos. São nesses repositórios que ficam armazenados o código fonte e os versionamentos deles.

O acesso e organização do código fonte dentro desses repositórios fica por responsabilidade da Gerência de Operações de TI que é responsável pela sustentação dos códigos implantados.

Os projetos deverão solicitar a equipe responsável pelo repositório para garantir os acessos aos códigos.

6.3.1. NOMENCLATURA DE BRANCH

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	8 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

A nomenclatura de branches nos repositórios deverá sempre seguir um padrão estipulado, com exceção das branches PROD e MAIN.

A Branch PROD deverá ser sempre relacionada a versão aplicada em produção, não podendo ser alterada até a entrega de um novo pacote no ambiente.

A Branch MAIN deverá ser sempre a versão base para geração das demais branches. Está branch também será a que irá receber os commits das demandas prontas e homologadas e que irão preparar as gestões de mudança para subida em produção.

A nomenclatura das branches de projetos deverá sempre começar por: [PROJ-NomeProjeto]. Se o projeto tiver mais de uma versão a ser lançado (em caso de um projeto tiver mais de uma fase) deverá ser feito o adicional ao final de 'Vx' onde 'x' se trata da numeração da fase (Exemplo: [PROJ-EXEMPLOV1] ou [PROJ-EXEMPLO]). O nome não poderá ter mais que 25 caracteres.

A nomenclatura de demandas de melhoria deverá começar por: [DEM-SIS-xxxx]-TítuloDemanda, o nome não poderá ter mais que 25 caracteres.

A nomenclatura de uma demanda de correção de erros deverá começar por: [BUGFIX]-NomeBug, o nome não poderá ter mais que 25 caracteres.

6.3.2. PERMISSÃO E ACESSO

A permissão de acesso deverá ser feita e liberada pela equipe de sustentação seguindo as diretrizes aqui definidas.

6.3.2.1. ANTIVIRUS

Todo colaborador que tiver acesso aos repositórios do Grupo CPFL deverá ter um software AntiVirus ativo e licenciado em sua máquina para que haja menor risco de sequestro de código fonte.

Essa informação deverá ser confirmada com print mostrando data e hora se o colaborador for de alguma empresa terceirizada.


6.3.2.2. MATRÍCULA CPFL

Todos os colaboradores que solicitarem acesso aos repositórios da CPFL deverão obrigatoriamente ter matrícula dentro do Active Directory.

6.3.2.2.1. SISTEMAS LEGADO

Para acesso ao código legado que estiver dentro do Github, deverá ser solicitado a inserção da matrícula do colaborador dentro de dois grupos de acesso no Active Directory, segue:

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	9 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

- **GitHub User**
- **GitHub – ‘projeto’**

6.3.2.2.2. NOVAS APLICAÇÕES

Para acesso ao código da nova aplicação em desenvolvimento que estiver dentro do Github, deverá ser solicitado a inserção da matrícula do colaborador dentro de dois grupos de acesso no Active Directory, segue:

- **GitHub User**
- **GitHub – ‘projeto’**

6.3.3. SISTEMAS LEGADO

Sistemas Legado do Grupo CPFL utilizam atualmente as seguintes ferramentas de versionamento e repositório de código:

- SourceSafe
- SVN Tortoise
- Microsoft Team Foundation Server
- Github

Os novos desenvolvimentos de código deverão respeitar a ferramenta de armazenamento, podendo ser alterados somente em caso de refatoramento ou arquivamento para histórico do código.

6.3.4. NOVOS SISTEMAS

Projetos que estejam criando novas aplicações que terão seu código fonte gerido e armazenado pelo Grupo CPFL deverão ser obrigatoriamente armazenados no Github da CPFL, devidamente organizado e padronizado.

6.4. FRAMEWORKS


Todo e qualquer Framework que for proposto para utilização em codificação para as soluções do Grupo CPFL deverão passar por análise da Gerência de Arquitetura e Dados (EIAD) da CPFL.

As avaliações deverão buscar os benefícios da utilização, como curva de aprendizagem, velocidade de entrega de código e customização.

Não será necessário a avaliação de Frameworks já utilizados em sistemas que já possuem sustentação pela Equipe de Operação de TI da CPFL e seus fornecedores.

6.5. CONECTORES DE BANCO DE DADOS

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	10 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

As aplicações do Grupo CPFL deverão trazer documentado a versão para utilização dos conectores de banco de dados.

Também deverá ser guardado em repositório todo e qualquer instrução e instalador para montagem do ambiente de desenvolvimento, visando eliminar empecilhos e atrasos a novos desenvolvedores.

6.6. PROGRAMAÇÃO EM BANCO DE DADOS

Diversas soluções da CPFL contêm hoje parte de sua lógica de negócio inserida dentro de trechos de código em seus bancos de dados.

Essa prática deve ser restringida e discutida ao máximo para que se siga o padrão de desenvolvimento, versionamento e estrutura de boas práticas do Grupo.

6.6.1. TRIGGERS

A criação de Triggers deverá ser utilizada sempre se voltando a ganho de performance da solução, e onde as regras de negócio não tenham margem para grandes alterações.

Também deverá estar documentado e vinculado ao repositório de código da solução para versionamento e rastreo na depuração de problemas de código.

6.6.2. PROCEDURES

Procedures não devem ser utilizados como opção primária para desenvolvimento de código fonte, devido a maior complexidade na manutenção, depuração e versionamento.

6.6.3. VIEWS


A Criação de Views para geração de relatórios ou retorno de informações para preenchimento de tela deverá ser analisada e acompanhada de um DBA, onde ele deverá gerar o custo de query da View. Essa atividade é necessária devido a possibilidade de impactos diretos no consumo do servidor associado ao Banco de Dados da aplicação.

Se for possível, sempre que houver necessidade de consulta de dados deverá ser avaliado a opção de uso do Data Lake da CPFL.

6.6.4. DBLINK

Conexões via DBLink (ou similares) entre bases de dados de soluções deverão ser avaliadas pela equipe de Arquitetura. As conexões diretas entre bases podem dificultar a análise e monitoramento das interfaces entre as soluções, sendo assim não deverão estar liberadas para uso indiscriminado.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	11 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

6.7. MACHINE LEARNING E INTELIGÊNCIA ARTIFICIAL

O Desenvolvimento de soluções utilizando Machine Learning e Inteligência Artificial deverá sempre ser analisado pela Gerência de Arquitetura e Dados (EIAD).

Haverá necessidade de análise dos projetos para verificar se os mesmos criarão algo do zero ou se já há disponibilidade de contratar serviços treinados de Machine Learning.

6.8. POWER PLATFORM

O Desenvolvimento de soluções utilizando Power Platform deverão seguir as diretrizes e boas práticas definidas pelo CoE de Power Platform.

Toda aplicação que seja de uso corporativo ou crítica deve possuir análise/liberação da estrutura do Centro de Excelência para início do desenvolvimento.

O mesmo reserva o direito de seguir as práticas de desenvolvimento em 3 camadas, DEV, QAS e PRD. A aplicação só poderá ser colocada em ambiente produtivo e utilizada pelo negócio após passar por testes e homologação.

Todo e qualquer liberação de estrutura dessa plataforma de Low Code também deverá passar pela mesma célula.

6.9. ANÁLISE DE CÓDIGO

6.9.1. CODE REVIEW E PULL REQUEST

Fica convencionado que toda alteração de código deverá passar por uma análise de código para avaliação. Esta avaliação deverá gerar evidências por fluxo automatizado (Continuous Integration / Continuous Deployment) ou por comunicações digitais (e-mail ou ferramentas de repositório).

6.9.2. SONAR QUBE


Fica convencionado que o Grupo CPFL exigirá também que ferramentas de análise de código como o SONAR QUBE sejam efetivamente utilizadas junto aos desenvolvedores, por meio de esteira automatizada ou como plug-in de IDEs de desenvolvimento.

O código será analisado baseado em métricas de mercado e boas práticas de protocolos.

6.9.3. ESTEIRA DEVSECOPS

Todo novo projeto de solução de TI que for hospedado em ambiente CPFL deverá ter como pré-requisito a criação de uma esteira DEVSECOPS.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	12 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Essa esteira DEVSECOPS deve estar de acordo com os padrões aplicados pela equipe da Gerência de Segurança da Informação, Gerencia de Infraestrutura e Operações de TI.

6.10. PROIBIÇÕES

A seguir estão as definições que ficam terminantemente proibidas nas entregas de código para o Grupo CPFL. O não cumprimento dessas normas acarretará em não aceitação das demandas e projetos até a adequação correta dos casos.

6.10.1. HARDCODE

Fica definido que o termo 'HARDCODE' está associado a escrita de informações diretamente no código fonte mas que deveriam estar gravados em algum lugar com possibilidades de alteração via sistema/parâmetros.

6.10.1.1. CÓDIGO SQL

Fica terminantemente proibido que códigos SQL recebam diretamente a concatenação de variáveis.

Deverá ser utilizado o padrão de passagem de parâmetros para as queries que estão escritas em código SQL dentro do código Fonte. Também está incluso nessa regra a utilização de variáveis diretamente em código SQL escritos a partir de estruturas semelhantes a Entity Framework que utilizem código SQL aberto como o exemplo abaixo:

```
using (var ctx = new SchoolDBEntities())
{
    var studentList = ctx.Students
        .SqlQuery("Select * from Students")
        .ToList<Student>();
}
```


6.10.1.2. CREDENCIAIS

Fica terminantemente proibido que códigos Credenciais de acesso de qualquer espécie seja diretamente escrito no código fonte das soluções do Grupo CPFL.

6.10.2. USO INDISCRIMINADO DE LISTAS EM MEMÓRIA

Fica terminantemente proibido o uso indiscriminado da criação e preenchimento de Listas em memória dentro de laços de repetição.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	13 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de Software

Esse tipo de prática impacta negativamente toda a infraestrutura planejada, os custos envolvidos e a experiência do cliente CPFL.

6.10.3. USO INDISCRIMINADO DE FULL TABLE SCAN

Fica terminantemente proibido a seleção geral dos dados de tabelas em que há sinalização de grande massa de dados ou possibilidade de crescimento exponencial de dados. Qualquer possibilidade de utilização de tal funcionalidade deverá ser acompanhada de um DBA para avaliação de custo de máquina ainda em período de homologação das funções do entregável.

Os impactos gerais referentes a esse tipo de atividade podem colocar em risco a saúde e a operação da solução assim como prejudicar outras atividades correlacionadas e que dependam de integração e interface com os dados da solução em questão.

6.10.4. CONSUMO INDISCRIMINADO DE CONSULTAS A INTERFACES

Fica terminantemente proibido o uso indiscriminado de utilização de interfaces sem a devida necessidade, principalmente onde a informação poder ser carregada em memória para utilização posterior.

Fica também proibido a consulta a interfaces de outras soluções dentro de laços de repetição, onde poderá gerar um volume desnecessário de interações de rede e infraestrutura que impactará todo o ambiente relacionado.

6.10.5. CRIAÇÃO DE VARIÁVEIS DENTRO DE LAÇOS DE REPETIÇÃO

Fica proibido a criação de variáveis dentro de laços de repetição, sendo orientado apenas a criação anterior das mesmas. Esse tipo de protocolo visa manter o consumo de memória da solução e a diminuir a complexidade do código fonte facilitando as documentações e comentários sobre as regras de negócio.


7. CONTROLE DE REGISTROS

Identificação	Armazenamento e Preservação	Proteção (acesso)	Recuperação e uso	Retenção	Disposição
(A)	(B)	(C)	(D)	(E)	(F)

(A) Identificação:

Nome do documento deverá ser, por padrão, vinculados ao nome da demanda e do projeto que estão atrelados. Demandas deverão incrementar a documentação técnica dos sistemas que tiverem alteração/melhoria. Projetos deverão conter em suas documentações de arquitetura os padrões utilizados juntamente com as definições técnicas. Exemplo: "DEM-SIS-0002000-Padiao de Arquitetura"

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	14 de 15

 Confidencialidade	Tipo de Documento:	Procedimento
	Área de Aplicação:	Tecnologia de Informação
	Título do Documento:	Boas Práticas de Arquitetura e Desenvolvimento de
	Software	

(B) **Armazenamento e Preservação:** Local de armazenamento deverá ser o repositório de sustentação do sistema caso o sistema já esteja implantado e para projetos deverá estar armazenado junto as demais documentações.

(C) **Proteção (acesso):** Os arquivos deverão estar protegidos por backup e scans de segurança. O acesso não deverá ser restrito para as equipes de TI que prestarão sustentação/planejamento ou para as equipes de projetos.

(D) **Recuperação e uso:** Recuperação deverá ser feita baseado nos últimos dois registros do arquivo e o uso deverá ser feito com base na última versão disponibilizada.

(E) **Retenção:** Tempo indefinido, os arquivos mais antigos poderão corroborar com entendimento de regras ao longo do tempo, sendo inviável a exclusão deste tipo de documento.

(F) **Disposição:** Não haverá disposição dos arquivos, serão armazenados indefinidamente.

8. ANEXOS

Sem anexos até o momento.

9. REGISTRO DE ALTERAÇÕES

9.1. Colaboradores

Empresa	Área	Nome
CPFL Paulista	EIAD – Gerência de Arquitetura e Dados	Angelo Augusto Simonato Sanches
CPFL Paulista	EID – Gerencia de Operações Digitais	Danilo Rothe Soares Lopes
RGE	EID – Gerencia de Operações Digitais	Samuel Flores
CPFL Paulista	EID – Gerencia de Operações Digitais	Isaias Sena Valença
CPFL Paulista	EID – Gerencia de Operações Digitais	Wellington Gabriel
CPFL Paulista	EID – Gerencia de Operações Digitais	Thiago Henrique Souza Domingues
CPFL Paulista	EIS – Coordenação de Segurança de TI BR	Alexandre Hiroshi Marchioli Fukaya

9.2. Alterações

Versão Anterior	Data da Versão Anterior	Alterações em relação à Versão Anterior
v0		Criação do documento.

N.Documento:	Categoria:	Versão:	Aprovado por:	Data Publicação:	Página:
19369	Instrução	1.0	DIEGO HENRIQUE PERISSATO	022/11/2022	15 de 15