



Ministério da Educação
Instituto Federal de Educação, Ciência e Tecnologia do Piauí – IFPI
Coordenação Geral do Projeto INOVATECH

NOTA TÉCNICA COMPLEMENTAR Nº 03/2026 – INOVATECH

*Processo Seletivo Simplificado – Residência em Inovação Tecnológica
Edital 30/2026 – GAB/REI/IFPI*

ASSUNTO: Ambiente de Realização da Prova Prática – Especificação Técnica Definitiva dos Projetos Base, Versionamento, Nomenclatura de Pastas de Entrega e Procedimento de Verificação de Integridade por Hash

1. OBJETO

Esta Nota Técnica tem por objetivo prestar informações complementares, de caráter definitivo e inequívoco, acerca do ambiente técnico de realização da Prova Prática do Processo Seletivo INOVATECH, respondendo aos questionamentos formalizados por candidatos após a publicação das Notas Técnicas n.º 01 e 02/2026.

São objeto desta Nota: (a) a especificação exata das versões de todos os pacotes e tecnologias presentes nos projetos base disponibilizados nos laboratórios; (b) a definição dos códigos e nomenclatura das pastas de entrega atribuídos a cada candidato; e (c) a descrição do procedimento de verificação de integridade por hash que será adotado para garantir transparência e auditabilidade do processo.

2. VERSÕES EXATAS DOS PROJETOS BASE

Em resposta aos questionamentos recebidos, a Coordenação Geral do Projeto INOVATECH informa que os projetos base estarão previamente instalados e configurados nos computadores dos laboratórios. Os candidatos não precisarão, e não poderão, instalar quaisquer pacotes ou dependências durante a realização da prova, dado que não haverá acesso à internet.

Todas as versões foram fixadas de forma exata (pinagem no formato X.Y.Z), garantindo que o ambiente seja idêntico em todos os computadores de todos os laboratórios. A tabela a seguir lista cada tecnologia e sua versão correspondente:

Camada	Tecnologia / Pacote	Versão Exata
Backend	Python	3.12.x (sistema)
Backend	Django	5.2.13 (LTS)

Backend	djangoestframework	3.17.1
Backend	djangoestframework-simplejwt	5.5.1
Backend	django-filter	25.2
Backend	django-cors-headers	4.9.0
Backend	drf-spectacular	0.29.0
Backend	asgiref	3.11.1
Backend	sqlparse	0.5.5
Backend	FastAPI	0.136.0
Backend	Uvicorn	0.45.0
Backend	Starlette	1.0.0
Backend	SQLModel	0.0.38
Backend	Alembic	1.18.4
Backend	SQLAlchemy	2.0.49
Backend	Mako	1.3.11
Backend	PyJWT	2.9.0
Backend	python-jose[cryptography]	3.5.0
Backend	passlib	1.7.4
Backend	bcrypt	3.2.2
Backend	python-multipart	0.0.26
Backend	email-validator	2.3.0
Backend	httpx	0.28.1
Backend	anyio	4.13.0
Backend	pydantic	2.13.3
Backend	pydantic-settings	2.14.0
Backend	annotated-types	0.7.0
Backend	Node.js	22.x (LTS – sistema)
Backend	Express	4.21.2
Backend	TypeORM	0.3.28
Backend	better-sqlite3	12.9.0
Backend	jsonwebtoken	9.0.3
Backend	bcryptjs	2.4.3
Backend	reflect-metadata	0.2.2
Backend	cors	2.8.6
Backend	dotenv	17.4.2
Backend	class-validator	0.15.1
Backend	class-transformer	0.5.1
Backend	TypeScript (Express)	5.8.3
Backend	ts-node	10.9.2

Backend	ts-node-dev	2.0.0
Backend	@types/express	4.17.21
Backend	@types/node	22.15.3
Backend	@types/jsonwebtoken	9.0.10
Backend	@types/bcryptjs	2.4.6
Backend	@types/cors	2.8.19
Backend	@types/better-sqlite3	7.6.13
Backend	rimraf	5.0.10
Frontend	Vite	5.4.14
Frontend	TypeScript (Vite templates)	5.6.3
Frontend	React	18.3.1
Frontend	@types/react	18.3.28
Frontend	@types/react-dom	18.3.7
Frontend	@vitejs/plugin-react	4.7.0
Frontend	react-router-dom	6.30.3
Frontend	axios	1.15.2
Banco de Dados	SQLite	3.x (nativo do sistema)

As versões foram selecionadas observando os seguintes critérios técnicos, que os candidatos devem ter ciência:

- Django: fixado na série 5.2.x LTS (Long-Term Support, com suporte até abril de 2028). A versão 6.0.x, embora mais recente, introduz quebras de compatibilidade (breaking changes) e não corresponde ao ecossistema majoritariamente estudado pelos candidatos.
- Express: fixado na versão 4.21.2. A versão 5.x, embora já lançada, apresenta quebras de compatibilidade relevantes no tratamento de erros e rotas assíncronas em relação à série 4.x, que é a amplamente coberta nos materiais de estudo e no conteúdo programático da prova.
- React: fixado na versão 18.3.1. A versão 19.x, lançada recentemente, introduz alterações no modelo de renderização e nas APIs de concorrência, com impacto para candidatos que estudaram React 18.
- TypeScript: o projeto Express utiliza TypeScript 5.8.3 (pinado explicitamente como devDependency). Os projetos de frontend utilizam TypeScript 5.6.3, que é a versão instalada pelos templates Vite 5.x. São duas versões distintas do mesmo pacote, cada uma no escopo de seu respectivo projeto, sem conflito entre si.
- Vite: o runtime vite está fixado na versão 5.4.14, compatível com o ecossistema React 18 e TypeScript 5.x. Importante: os pacotes vite e create-vite possuem numerações de versão independentes no npm. O scaffolding utiliza create-vite@5.5.5 (última versão publicada da série 5.x do scaffolding), enquanto o runtime vite@5.4.14 é instalado em seguida, sobrescrevendo a versão padrão gerada pelo scaffold. Do ponto de vista do candidato, o ambiente de desenvolvimento corresponde ao Vite 5.4.14.

- PyJWT e python-jose: ambas as bibliotecas de autenticação JWT estão disponíveis no projeto FastAPI. O PyJWT NÃO é instalado automaticamente pelo FastAPI e, portanto, foi incluído explicitamente. O python-jose é instalado com o extra [cryptography], o que habilita algoritmos assimétricos (RS256, ES256, etc.) além dos simétricos (HS256). O candidato poderá utilizar qualquer uma das duas bibliotecas conforme sua preferência.
- passlib e bcrypt: disponíveis para hashing de senhas no projeto FastAPI. Também não são dependências automáticas do FastAPI e foram incluídas explicitamente.
- python-multipart: obrigatório para uso do OAuth2PasswordRequestForm do FastAPI. Incluído explicitamente.

Não haverá disponibilização de pacotes além dos listados nesta Nota Técnica. O candidato deve planejar sua solução considerando exclusivamente as bibliotecas aqui especificadas.

3. NOMENCLATURA DAS PASTAS DE ENTREGA

Nos computadores dos laboratórios haverá uma pasta de entrega previamente criada, localizada em:

`~/inovatech/entrega`

Ao iniciar a prova, o candidato deverá renomear essa pasta para o formato abaixo, substituindo ## pelo seu código de dois dígitos com zero à esquerda quando necessário (01 a 40):

`entrega_##`

O código de cada candidato é determinado pela sua ordem de classificação na Prova Objetiva, conforme tabela a seguir. O comando para renomear a pasta, a ser executado no terminal a partir de ~/inovatech, é:

`mv entrega entrega_##`

Código	Nome	Tipo de Vaga	Pontos
entrega_01	Ruan Macedo Santos	Reservada IFPI	30
entrega_02	Gabriel Augusto de Oliveira Silva	Ampla	29
entrega_03	Halyson Itallo Cunha Pimentel	Reservada IFPI	28
entrega_04	Ryan Faustino Carvalho	Reservada IFPI	28
entrega_05	João Pedro de Sousa Rodrigues	Reservada IFPI	27
entrega_06	Raffael Ferreira Fernandes	Ampla	27
entrega_07	Artur Vinicius Silva Sousa	Ampla	27
entrega_08	Saulo Pietro Lacerda dos Santos	Ampla	27

entrega_09	Nihl Deyves de Miranda Piauilino	Ampla	26
entrega_10	Gabriel Costa Souza	Reservada IFPI	26
entrega_11	Wesley Vitor Barbosa Brandão	Ampla	26
entrega_12	Francisco Alexandro Conceição Sousa	Reservada IFPI	26
entrega_13	Mayara dos Santos Gregório	Ampla	26
entrega_14	Bianca Almeida de Oliveira Bezerra	Reservada IFPI	26
entrega_15	Luis Henrique de Moura Santos	Reservada IFPI	26
entrega_16	Xamã Cardoso Mendes Santos	Reservada IFPI	26
entrega_17	João Paulo Lopes da Silva	Reservada IFPI	26
entrega_18	Joao Carlos Cunha da Silva	Ampla	25
entrega_19	Marcos Valdecy Macedo Costa Leite	Ampla	25
entrega_20	Hermínio de Barros e Silva Neto	Reservada IFPI	25
entrega_21	Renan de Almeida Lira	Reservada IFPI	25
entrega_22	Wesley de Jesus Souza Morais	Ampla	25
entrega_23	Victor José Pires Sena	Ampla	25
entrega_24	Murilo Pereira Monte	Ampla	25
entrega_25	Marcelo Garcia Felipe	Reservada IFPI	24
entrega_26	Benjamim de Pinho Sabino	Ampla	24
entrega_27	Rômulo de Sousa Carvalho	Reservada IFPI	24
entrega_28	Mateus Rodrigues de Sousa Borja	Ampla	24
entrega_29	Israel Benvindo Pereira	Reservada IFPI	24
entrega_30	Pedro Henrique Vieira	Reservada IFPI	24
entrega_31	Camilly Larissa Costa Silva	Ampla	24
entrega_32	Diego de Sousa Araújo	Reservada IFPI	24
entrega_33	Nicole Ellen Magalhães Silvestre	Ampla	24
entrega_34	José Guilherme Rodrigues de Sousa	Reservada IFPI	24
entrega_35	Maria Ingrid Xavier Sousa	Ampla	24
entrega_36	Sergio Guthyerres dos Santos de Morais	Ampla	24
entrega_37	Delwin Silva Menor	Reservada IFPI	24
entrega_38	Luiz Fernando Mendes Osorio	Ampla	23
entrega_39	João Marcos Costa Ferreira	Ampla	23
entrega_40	Paulo Bruno Fernandes Barros Noieto	Ampla	23

Ao término da prova, o candidato deverá certificar-se de que seu projeto desenvolvido está integralmente depositado dentro da pasta que lhe foi atribuída. Não será permitida a entrega em outro local. A solução presente na pasta no momento do encerramento da prova será a única considerada para avaliação.

4. PROCEDIMENTO DE VERIFICAÇÃO DE INTEGRIDADE POR HASH

4.1 Objetivo

A fim de garantir transparência, rastreabilidade e isonomia a todos os participantes, a Coordenação adotará um sistema de verificação de integridade baseado em hash criptográfico SHA-256, aplicado em dois momentos distintos: antes do início da prova (selagem dos projetos base) e ao término da prova (registro da entrega de cada candidato).

4.2 Hash Público dos Projetos Base

Antes do início da prova, a Coordenação executará um procedimento de selagem que percorre todos os arquivos dos projetos base disponibilizados, calcula o hash SHA-256 de cada arquivo individualmente e, a partir da concatenação ordenada desses hashes, gera um único Hash Raiz representativo do estado integral do ambiente. Esse Hash Raiz será o Hash Público Oficial dos Projetos Base.

O Hash Público Oficial será divulgado a todos os candidatos presentes no início da prova, por meio de sua exibição nos computadores dos laboratórios e, posteriormente, publicado no site do processo seletivo. Qualquer candidato poderá, com o Hash Público em mãos, verificar que a pasta de projetos base em seu computador é bit a bit idêntica à dos demais.

4.3 Comprovante de Entrega do Candidato

Ao concluir sua solução e antes do encerramento oficial da prova, o candidato deverá executar o comando de geração de comprovante pelo terminal. O procedimento é o seguinte:

- i) Abrir o terminal no computador do laboratório.
- ii) Navegar até o diretório ~/inovatech com o comando: `cd ~/inovatech`
- iii) Executar o comando: `inovatech-submit`
- iv) Informar o número de inscrição e o nome completo quando solicitado pelo sistema.
- v) O sistema calculará o Hash SHA-256 de todos os arquivos presentes na pasta de entrega do candidato e exibirá o Hash da Entrega na tela.
- vi) O candidato deverá anotar ou fotografar o Hash da Entrega exibido. Um arquivo de comprovante será automaticamente salvo dentro da pasta de entrega.

O candidato que não executar o procedimento de geração do comprovante não terá prejuízo na avaliação, pois a pasta de entrega será igualmente avaliada. Contudo, a ausência do comprovante inviabilizará eventual contestação de integridade posterior.

4.4 Verificação Pós-Resultado

Após a divulgação do resultado da Prova Prática, o candidato que desejar verificar se a pasta avaliada corresponde exatamente ao que foi entregue poderá:

- vii) Solicitar à Coordenação, pelo endereço eletrônico inovatech@ifpi.edu.br, o Hash da pasta avaliada referente à sua inscrição.
- viii) Comparar o Hash fornecido pela Coordenação com o Hash registrado em seu comprovante de entrega.
- ix) Se os dois hashes forem idênticos, o conteúdo avaliado é exatamente o que foi entregue. Qualquer divergência deverá ser comunicada à Coordenação no prazo recursal previsto no Edital 30/2026.

4.5 Metodologia de Cálculo do Hash

Para fins de transparência e possibilidade de auditoria independente, a metodologia de cálculo adotada é a seguinte:

- x) São considerados apenas os arquivos de código-fonte e configuração relevantes, excluindo-se automaticamente: diretórios de dependências instaladas (.venv/, node_modules/), arquivos compilados (__pycache__/, dist/, *.pyc, *.pyo), arquivos de log e arquivos de sistema (.DS_Store).
- xi) Os arquivos são listados em ordem lexicográfica determinística (find + sort), garantindo que o mesmo conteúdo produza sempre o mesmo hash, independentemente da máquina ou data de execução.
- xii) O Hash SHA-256 de cada arquivo é calculado individualmente.
- xiii) Os hashes individuais são concatenados na ordem determinada pelo passo ii e submetidos a um último cálculo SHA-256, resultando no Hash Raiz final.

5. DISPOSIÇÕES FINAIS

5.1. As informações contidas nesta Nota Técnica são definitivas e complementam as Notas Técnicas n.º 01 e 02/2026. Em caso de conflito entre os documentos, prevalece o disposto na nota de publicação mais recente.

5.2. Não serão respondidos questionamentos cujas respostas já constem desta ou das demais Notas Técnicas publicadas.

5.3. Dúvidas remanescentes sobre o processo seletivo devem ser encaminhadas exclusivamente para o endereço eletrônico inovatech@ifpi.edu.br, conforme previsto no Edital 30/2026.

Teresina, Piauí, 21 de abril de 2026.

Coordenação Geral do Projeto INOVATECH

Instituto Federal de Educação, Ciência e Tecnologia do Piauí – IFPI