



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Tecnologia em Telemática

**DESENVOLVIMENTO DE UM SISTEMA WEB
PARA GERENCIAMENTO DE HORÁRIOS
ESCOLARES - GHE**

MATEUS SEBASTIAN DE ALBUQUERQUE ROCHA

Orientador: Dr. Éwerton Rômulo Silva Castro

Campina Grande, 2026

Mateus Sebastian de Albuquerque Rocha



Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Tecnologia em Telemática

DESENVOLVIMENTO DE UM SISTEMA WEB PARA GERENCIAMENTO DE HORÁRIOS ESCOLARES - GHE

MATEUS SEBASTIAN DE ALBUQUERQUE ROCHA

Monografia apresentada à Coordenação do
Curso de Telemática do IFPB - Campus
Campina Grande, como requisito parcial
para conclusão do curso de Tecnologia em
Telemática.

Orientador: Dr. Éwerton Rômulo Silva Castro

Campina Grande, 2026

Catlogação na fonte:
Ficha catalográfica elaborada por Gustavo César Nogueira da Costa – CRB 15/479

R672d Rocha, Mateus Sebastian de Albuquerque
Desenvolvimento de um sistema web para gerenciamento
de horários escolares – GHE / Mateus Sebastian de
Albuquerque Rocha. - Campina Grande, 2026.

51 f.: il.

Trabalho de Conclusão de Curso (Graduação em
Tecnologia em Telemática) - Instituto Federal da Paraíba,
2026.

Orientador: Prof.Dr. Éwerton Rômulo Silva Castro

1. Telemática. 2. Gestão escolar. 3. Desenvolvimento de
sistemas - sistema web. 4. Automação de processos
acadêmicos - gestão de horários. I. Castro, Éwerton
Rômulo Silva. II Título.

CDU 004:37

DESENVOLVIMENTO DE UM SISTEMA WEB PARA GERENCIAMENTO DE HORÁRIOS ESCOLARES - GHE

MATEUS SEBASTIAN DE ALBUQUERQUE ROCHA

Dr. Éwerton Rômulo Silva Castro
Orientador

Dr. Anderson F. B. F. da Costa
Membro da Banca

Marcos V. C. M. Andrade
Membro da Banca

Campina Grande, Paraíba, Brasil
Junho/2026

Dedico aos meus pais e minha noiva.

I'm the architect of my own destruction...
Prince of Persia (PoP: Warrior Within)

Agradecimentos

Agradeço aos meus pais, pelo apoio constante durante minha formação e pela confiança nos momentos de maior dificuldade. Agradeço também à minha noiva, pela paciência, incentivo e presença ao longo do desenvolvimento deste trabalho.

Ao meu orientador, professor Dr. Éwerton Rômulo Silva Castro, agradeço pela orientação, pelas revisões e pelas contribuições que ajudaram a transformar o projeto em um trabalho acadêmico mais consistente. Agradeço ainda aos professores do Curso Superior de Tecnologia em Telemática do IFPB, Campus Campina Grande, pelos conhecimentos compartilhados durante a graduação.

Por fim, agradeço aos colegas e a todos que, direta ou indiretamente, contribuíram para a conclusão deste trabalho.

Resumo

Este trabalho apresenta o desenvolvimento do Gerenciador de Horários Escolares (GHE), sistema *web* voltado à consulta pública, coleta de indisponibilidade docente e geração automática de grades acadêmicas. O recorte inicial adotado foi o Instituto Federal da Paraíba (IFPB), Campus Campina Grande. A metodologia combinou levantamento de requisitos e regras de negócio, modelagem incremental com Django e PostgreSQL, uso do Supabase como banco gerenciado, extração de dados reais do portal público de horários do IFPB com *Requests* e *Beautiful Soup*, importação por etapas para o banco e validação funcional por meio de testes automatizados e inspeção das interfaces. Como resultados, o projeto disponibiliza navegação pública por cursos, períodos, professores, salas e laboratórios, matriz curricular por curso e período, estrutura multi-instituição e multi-campus, coleta restrita de indisponibilidade docente por enlace com identificador de acesso, área operacional da secretaria, geração de rascunhos, validação de conflitos, publicação da grade e edição posterior dos horários publicados. A base analisada reúne 1 instituição, 1 campus, 23 cursos, 95 períodos, 174 professores, 783 disciplinas, 793 vínculos entre curso, período e disciplina, 35 salas, 34 laboratórios e 2156 horários publicados no recorte inicial importado do campus. Em uma execução local com o curso de Telemática no semestre 2026.1, o gerador produziu 148 aulas, validou o rascunho sem pendências e publicou a grade substituindo os registros anteriores do mesmo curso e semestre. Os resultados obtidos indicam a viabilidade da arquitetura proposta e consolidam uma base funcional para evolução em exportações, notificações, otimização e integração futura com o SUAP.

Palavras-chave: horários escolares; geração automática; Django; IFPB.

Abstract

This work presents the development of the Gerenciador de Horários Escolares (GHE), a web system designed for public timetable consultation, teacher unavailability collection, and automatic generation of academic timetables. The initial institutional scope was the Federal Institute of Paraíba (IFPB), Campina Grande campus. The adopted methodology combined requirement and business rule analysis, incremental modeling with Django and PostgreSQL, the use of Supabase as a managed database, extraction of real timetable data from the IFPB public portal using Requests and BeautifulSoup, stepwise data import into the database, and functional validation through automated tests and interface inspection. As results, the project provides public navigation by course, academic period, professor, classroom and laboratory, a curriculum matrix per course and period, a multi-institution and multi-campus structure, restricted collection of teacher unavailability through individual access identifiers, an operational secretariat area, draft generation, conflict validation, timetable publication, and later editing of published schedules. The analyzed dataset contains 1 institution, 1 campus, 23 courses, 95 academic periods, 174 professors, 783 disciplines, 793 course-period-discipline relations, 35 rooms, 34 laboratories and 2156 published timetable entries in the initial imported campus scope. In a local execution with the Telematics course for the 2026.1 term, the generator produced 148 lessons, validated the draft without issues, and published the timetable by replacing previous records for the same course and term. The obtained results indicate the viability of the proposed architecture and establish a functional foundation for future stages involving exports, notifications, optimization, and integration with SUAP.

Keywords: school timetabling; automatic generation; Django; IFPB.

Sumário

Lista de Abreviaturas	xi
Lista de Figuras	xii
Lista de Tabelas	xiii
1 Introdução	1
1.1 Justificativa e Relevância do Trabalho	2
1.2 Objetivos	3
1.2.1 Objetivo Geral	3
1.2.2 Objetivos Específicos	3
1.3 Metodologia	4
1.4 Organização do Documento	4
2 Fundamentação Teórica e Tecnologias Utilizadas	6
2.1 <i>Timetabling</i> educacional e restrições acadêmicas	6
2.2 Sistemas acadêmicos e organização por instituição e campus	7
2.3 Tecnologias adotadas no GHE	7
2.3.1 Django e organização da aplicação	7
2.3.2 PostgreSQL, Supabase e integridade relacional	8
2.3.3 Django Templates, HTML, CSS e JavaScript	8
2.3.4 Requests e Beautiful Soup na coleta de dados	9
2.3.5 Configuração de ambiente e implantação	9
2.3.6 Segurança web como requisito transversal	9
3 Desenvolvimento do GHE	11
3.1 Requisitos e regras de negócio	11
3.2 Casos de uso do sistema	12
3.3 Arquitetura modular do sistema	13
3.4 Modelo de dados principal	14
3.4.1 Modelagem do banco de dados	14
3.4.2 Modelo relacional do banco de dados	15
3.5 Segurança de dados e segurança web	17

3.6	Fluxo de coleta e importação dos dados	18
3.7	Módulos funcionais implementados	19
3.7.1	Consulta pública de horários	19
3.7.2	Matriz curricular	20
3.7.3	Coleta de indisponibilidade docente	20
3.7.4	Área da secretaria	21
3.7.5	Motor de geração, validação e publicação	25
4	Resultados Obtidos	28
4.1	Base institucional e acadêmica importada	28
4.2	Consulta pública e organização da grade	29
4.3	Coleta e gestão de indisponibilidade docente	29
4.4	Geração automática de horários	29
4.5	Operação da secretaria	30
4.6	Validação técnica	30
4.7	Limitações atuais	31
5	Considerações Finais e Sugestões para Trabalhos Futuros	32
5.1	Sugestões para Trabalhos Futuros	32
	Referências Bibliográficas	34
A	Estrutura resumida do projeto GHE	36
A.1	Aplicações principais	36
A.2	Rotas públicas e restritas implementadas	36
A.3	Recorte do modelo relacional	37
A.4	Relações centrais do modelo	37
A.5	Casos de uso por ator	37
A.6	Comandos relevantes	37

Lista de Abreviaturas

API	<i>Application Programming Interface</i> - Interface de Programação de Aplicações
CSS	<i>Cascading Style Sheets</i> - Folhas de Estilo em Cascata
CSV	<i>Comma-Separated Values</i> - Valores Separados por Vírgula
DTL	<i>Django Template Language</i> - Linguagem de <i>templates</i> do Django
GHE	Gerenciador de Horários Escolares
HTML	<i>HyperText Markup Language</i> - Linguagem de Marcação de Hipertexto
IFPB	Instituto Federal da Paraíba
ORM	<i>Object-Relational Mapping</i> - Mapeamento Objeto-Relacional
PDF	<i>Portable Document Format</i> - Formato Portátil de Documento
SQL	<i>Structured Query Language</i> - Linguagem de Consulta Estruturada
SSL	<i>Secure Sockets Layer</i> - Camada de Soquetes Seguros
SUAP	Sistema Unificado de Administração Pública

Lista de Figuras

3.1	Diagrama geral de casos de uso do GHE.	13
3.2	Casos de uso do fluxo de geração, validação e publicação pela secretaria. . .	13
3.3	Exemplo simplificado da modelagem relacional usada para estruturar a base acadêmica, as restrições de entrada, os rascunhos e a grade publicada. . . .	15
3.4	Exemplo de modelo relacional do banco de dados do GHE com entidades, atributos principais, cardinalidades e relações operacionais de geração. . . .	17
3.5	Central de navegação de horários do GHE, organizada por modalidade, curso e período.	19
3.6	Grade semanal pública do curso de Telemática no GHE.	20
3.7	Página dedicada de professor com refinamento contextual por cursos, salas e laboratórios.	21
3.8	Matriz curricular pública do curso de Telemática no GHE.	22
3.9	Edição de indisponibilidade docente pela área da secretaria.	22
3.10	Painel da secretaria com indicadores de geração, validação e publicação. . .	23
3.11	Tela de vínculos professor-disciplina usada como entrada do gerador.	24
3.12	Configuração de horários permitidos por curso e semestre.	24
3.13	Prévia do rascunho gerado antes da publicação na consulta pública.	25
3.14	Validação do rascunho antes da publicação da grade.	26
3.15	Edição da grade publicada pela secretaria após a publicação do rascunho. . .	26
3.16	Horário gerado para o 6º período de Telemática após a execução da geração.	27

Lista de Tabelas

2.1	Tecnologias utilizadas no GHE e seu papel no projeto.	10
3.1	Requisitos funcionais do recorte final do GHE.	11
3.2	Síntese das aplicações do GHE e suas responsabilidades.	14
4.1	Métricas previstas e realizadas da base importada no GHE.	28
4.2	Resultado de geração e publicação para o curso de Telemática no semestre 2026.1.	30
4.3	Resumo das principais verificações técnicas realizadas no GHE.	31
A.1	Recorte textual do modelo relacional do GHE.	38
A.2	Relações centrais do modelo relacional do GHE.	38
A.3	Resumo dos casos de uso por ator no GHE.	39

Capítulo 1

Introdução

A organização de horários acadêmicos é uma atividade recorrente nas instituições de ensino e envolve a conciliação de recursos, restrições e necessidades de diferentes atores. No contexto do ensino superior e da educação profissional, o problema deixa de ser apenas operacional e passa a depender também de critérios institucionais, disponibilidade docente, uso de salas e laboratórios, oferta por período letivo e mecanismos de divulgação para a comunidade acadêmica. A literatura de *timetabling* educacional (alocação de horários acadêmicos) mostra que esse tipo de problema combina restrições duras, que não podem ser violadas, e restrições suaves, cujo atendimento melhora a qualidade final da grade [Ceschia, Gaspero e Schaerf 2022; Müller, Rudová e Müllerová 2025].

No IFPB, Campus Campina Grande, a publicação dos horários acadêmicos já ocorre por meio do portal público de horários do campus, disponível em `horarios.ifpb.edu.br` [Instituto Federal da Paraíba 2026]. Esse portal cumpre a função de divulgar a grade, mas não resolve integralmente outras demandas associadas ao gerenciamento do processo: navegação estruturada por diferentes recortes, manutenção de uma base relacional reutilizável, registro de indisponibilidade docente, configuração de regras de geração, produção de rascunhos, validação de conflitos e publicação controlada de novas grades.

Nesse cenário, o Gerenciador de Horários Escolares (GHE) foi desenvolvido como uma aplicação *web* (acessada pelo navegador) voltada ao gerenciamento de horários acadêmicos. O sistema tem foco inicial no IFPB, Campus Campina Grande, mas foi modelado desde o início para suportar múltiplas instituições e múltiplos campi. O projeto utiliza Django [Django Software Foundation 2025], PostgreSQL [The PostgreSQL Global Development Group 2025] e *Django Templates* (modelos de página do Django) [Django Software Foundation 2025], sem adoção de *frameworks* (estruturas de desenvolvimento) de aplicação de página única, e combina consulta pública, coleta de indisponibilidade e uma área restrita para a secretaria operar a geração automática.

O GHE incorpora uma etapa de coleta e normalização de dados reais do portal público, transforma esses dados em uma base acadêmica estruturada e implementa um fluxo de geração determinística de horários. Esse fluxo permite configurar vínculos professor-disciplina,

definir horários permitidos por curso e semestre, gerar uma grade em rascunho, validar conflitos e publicar a grade oficial que passa a alimentar a consulta pública. Assim, o trabalho documenta não apenas uma interface de visualização, mas a construção de uma base operacional para gerenciar o ciclo completo da grade horária.

Sob essa perspectiva, este trabalho contribui com a descrição do processo de construção de um sistema acadêmico orientado por requisitos reais, modelagem de dados consistente e evolução modular. Também documenta as decisões técnicas necessárias para transformar uma grade publicada em uma base navegável, validável, editável e reutilizável na geração automática de horários. A discussão é relevante porque estudos sobre *timetabling* frequentemente enfatizam algoritmos e *benchmarks* (bases ou cenários de comparação), enquanto a prática institucional demanda também infraestrutura de dados, interfaces públicas, mecanismos de coleta de restrições e ferramentas operacionais para a secretaria [Vrieling *et al.* 2019; Ceschia, Gaspero e Schaerf 2022].

1.1 Justificativa e Relevância do Trabalho

A existência de um portal público de horários não elimina a necessidade de um sistema de gerenciamento. A publicação final da grade representa apenas uma parte do processo institucional. Antes da divulgação, é necessário organizar cursos, períodos, disciplinas, professores, salas, laboratórios, semestres letivos, indisponibilidades e regras de alocação. Depois da publicação, também é necessário editar correções pontuais, manter rastreabilidade das alterações e garantir que a consulta pública reflita o semestre letivo correto.

Na prática, a motivação do GHE surgiu de dores observadas no uso cotidiano da grade: dificuldade de consultar o horário por diferentes recortes, ausência de uma base relacional reutilizável para análises, falta de visibilidade imediata sobre disciplinas ainda sem professor vinculado, coleta manual de indisponibilidades docentes e necessidade de conferir conflitos antes da publicação. Embora o horário já exista e seja publicizado, essas atividades continuam ocorrendo como etapas de bastidor e dependem de controle operacional além da simples visualização da grade final.

No recorte do IFPB, Campus Campina Grande, o portal público existente apresenta a grade à comunidade, mas não oferece, por si só, uma base preparada para coleta privada de indisponibilidade docente, configuração de horários liberados por curso, geração de rascunhos e validação das restrições antes da publicação. Essas lacunas justificam o desenvolvimento do GHE como uma camada de gerenciamento e evolução, e não como simples duplicação de uma página já existente.

A relevância técnica do trabalho está em estruturar uma solução que aproxima dados reais, regras de negócio e operação institucional. A geração automática de horários depende de uma base bem modelada, com vínculos explícitos entre campus, curso, período, disciplina, professor e espaço físico. Também depende de controles que impeçam conflitos de professor,

turma, sala e laboratório, além de respeitar indisponibilidades docentes e horários permitidos. O GHE aborda essas necessidades por meio de uma arquitetura modular e de um fluxo de geração e publicação integrado à consulta pública.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver e validar o Gerenciador de Horários Escolares (GHE) como um sistema *web* para consulta pública, coleta de indisponibilidade docente e geração automática de grades acadêmicas, tendo como foco inicial o IFPB, Campus Campina Grande.

1.2.2 Objetivos Específicos

- Modelar a base institucional e acadêmica do sistema com suporte a múltiplas instituições e múltiplos campi;
- Estruturar cursos, períodos, disciplinas, professores, salas, laboratórios, semestres letivos e horários com vínculo obrigatório ao campus;
- Coletar e importar para o banco de dados a grade pública atualmente utilizada pelo IFPB, Campus Campina Grande;
- Implementar navegação pública por curso, período, professor, sala e laboratório;
- Implementar a visualização de matriz curricular com vínculo explícito entre curso, período e disciplina;
- Implementar o fluxo restrito de coleta de indisponibilidade docente por enlace individual com identificador de acesso;
- Implementar uma área de secretaria para configurar vínculos professor-disciplina, horários permitidos, indisponibilidades e semestres ativos;
- Implementar a geração automática de rascunhos de grade a partir de regras determinísticas e restrições acadêmicas;
- Validar rascunhos de grade contra conflitos de professor, turma, sala, laboratório, horários permitidos, indisponibilidade docente e carga semanal;
- Publicar rascunhos validados como grade oficial do curso e permitir ajustes posteriores nos horários publicados;
- Validar tecnicamente o estado atual do projeto por meio de checagens, testes automatizados e inspeção funcional das interfaces.

1.3 Metodologia

A metodologia adotada neste trabalho é incremental. O processo iniciou-se com a definição de requisitos funcionais, requisitos não funcionais e regras de negócio, organizados no próprio repositório do projeto para orientar a implementação das fases previstas. Esses documentos definiram o escopo da consulta pública, da coleta de indisponibilidade, da estrutura multi-instituição e da geração automática.

Em seguida, foi realizada a modelagem da solução utilizando Django como *framework web* [Django Software Foundation 2025] e PostgreSQL como banco de dados relacional [The PostgreSQL Global Development Group 2025], com uso do Supabase como serviço gerenciado [Supabase 2025]. A estrutura do sistema foi dividida em módulos independentes por responsabilidade, separando contexto institucional, base acadêmica, horários, indisponibilidade docente, páginas públicas, secretaria, exportações e notificações.

Para aproximar o sistema da realidade do campus, foi implementada uma etapa de coleta automatizada dos dados do portal público de horários do IFPB com uso de *Requests* [Requests contributors 2025], biblioteca para requisições HTTP (*HyperText Transfer Protocol*, protocolo de transferência de hipertexto), e *Beautiful Soup* [Richardson 2025], biblioteca para leitura e extração de conteúdo HTML (*HyperText Markup Language*, linguagem de marcação de hipertexto). Os dados extraídos foram exportados para CSV (*Comma-Separated Values*, valores separados por vírgula) e posteriormente importados em etapas para o banco. Essa importação por etapas é idempotente, isto é, pode ser repetida sem duplicar os registros já existentes, e inclui tratamento de ambiguidades e geração de arquivos de revisão.

Após a consolidação da base acadêmica, foi implementado o fluxo de geração automática. A secretaria define os vínculos professor-disciplina do semestre, configura os horários permitidos do curso, revisa a indisponibilidade docente e aciona a geração de um rascunho. O sistema aloca aulas em blocos consecutivos, verifica restrições e registra o resultado em uma execução de geração. O rascunho pode ser revisado visualmente, validado e publicado como grade oficial.

Por fim, a validação considerou a execução de checagens do Django, testes automatizados e inspeção funcional das interfaces implementadas. Essa validação concentrou-se nas funcionalidades já entregues: consulta pública, matriz curricular, coleta de indisponibilidade, operação da secretaria, geração de rascunho, validação, publicação e edição da grade publicada.

1.4 Organização do Documento

No Capítulo 2, são apresentados os fundamentos do problema de *timetabling* educacional e as tecnologias utilizadas no desenvolvimento do GHE.

No Capítulo 3, é descrito o desenvolvimento do sistema, incluindo requisitos funcionais, casos de uso, segurança de dados, arquitetura modular, modelo de dados, fluxo de importa-

ção, módulos públicos, coleta de indisponibilidade e motor de geração automática.

No Capítulo 4, são discutidos os resultados obtidos com o projeto, contemplando a base importada, as interfaces públicas, a área da secretaria, a geração de rascunhos, a publicação da grade e a validação técnica realizada.

No Capítulo 5, são apresentadas as considerações finais e as sugestões para trabalhos futuros.

No Apêndice A, é apresentada uma visão resumida da organização do projeto e de seus principais componentes, incluindo um recorte do modelo relacional utilizado pelo GHE.

Capítulo 2

Fundamentação Teórica e Tecnologias Utilizadas

2.1 *Timetabling* educacional e restrições acadêmicas

O problema de *timetabling* educacional pode ser entendido como a alocação de eventos acadêmicos em faixas de tempo e recursos disponíveis, considerando simultaneamente um conjunto amplo de restrições. Em ambientes reais, essa atividade precisa respeitar conflitos de professor, uso exclusivo de salas e laboratórios, distribuição por períodos letivos, disponibilidade de espaços especializados e critérios institucionais de qualidade da grade [Ceschia, Gaspero e Schaerf 2022; Müller, Rudová e Müllerová 2025].

Na literatura, é comum distinguir entre restrições duras e restrições suaves. As restrições duras representam condições obrigatórias para que a grade seja viável, como a impossibilidade de um professor ministrar duas aulas no mesmo horário ou de uma mesma sala ser alocada para duas turmas simultaneamente. Já as restrições suaves estão relacionadas à qualidade da solução, por exemplo a distribuição mais equilibrada das aulas ao longo da semana ou a preferência por determinados espaços [Gaspero, McCollum e Schaerf 2007; Ceschia, Gaspero e Schaerf 2022].

No GHE, essa distinção foi traduzida para regras operacionais do sistema. O gerador trata como restrições obrigatórias os conflitos de professor, período ou turma, sala, laboratório, indisponibilidade docente e horários não liberados para o curso. A organização em pares de aulas consecutivas também foi tratada como regra de viabilidade do rascunho, com exceção configurável para disciplinas que podem ocupar quatro aulas consecutivas. Critérios como compactação da grade e aproveitamento de espaços são considerados dentro das decisões determinísticas do algoritmo, mas permanecem subordinados às restrições obrigatórias.

Além da formulação teórica do problema, estudos sobre a prática institucional mostram que a construção de horários em instituições de ensino superior envolve características locais, políticas acadêmicas, limitações físicas e formas específicas de governança. Por isso, a diferença entre soluções descritas em *benchmarks* e soluções efetivamente utilizadas nas

instituições é um aspecto importante para a engenharia de sistemas voltados a esse domínio [Vrieling *et al.* 2019]. No caso do GHE, essa constatação reforça a decisão de integrar modelagem de dados, interfaces operacionais e geração automática em um mesmo fluxo.

2.2 Sistemas acadêmicos e organização por instituição e campus

Uma dificuldade frequente em sistemas acadêmicos é representar corretamente a hierarquia institucional e as particularidades locais de cada unidade. Em instituições multicampi, cursos, professores, salas, laboratórios e horários não podem ser tratados como entidades globais sem contexto, pois cada campus possui sua própria oferta de cursos, seus próprios espaços e suas próprias regras operacionais.

O GHE foi modelado desde o início para suportar múltiplas instituições e múltiplos campi, mesmo tendo como foco inicial o IFPB, Campus Campina Grande. Essa decisão de projeto evita acoplamento indevido entre entidades de campi diferentes e preserva a coerência dos dados. Também prepara uma base relacional mais consistente para uma futura integração com sistemas institucionais, como o SUAP.

Outro ponto relevante é a distinção entre publicação da grade e gerenciamento da base que a sustenta. O portal público de horários do IFPB [Instituto Federal da Paraíba 2026] já fornece uma visão consolidada da grade para consulta. Entretanto, a construção do GHE procura ir além da exibição da grade, preparando uma base reutilizável para filtros, navegação por diferentes recortes, registro de indisponibilidade docente, geração automática de rascunhos e publicação controlada de novas grades.

2.3 Tecnologias adotadas no GHE

Do ponto de vista tecnológico, o GHE foi implementado com uma pilha orientada à simplicidade operacional, modularidade e facilidade de evolução. Como o objetivo do trabalho era construir um sistema aplicável ao contexto institucional, as tecnologias foram escolhidas para favorecer uma base relacional consistente, interfaces *web* tradicionais, fluxo de autenticação restrita para a secretaria, testes automatizados e implantação em ambiente externo.

2.3.1 Django e organização da aplicação

O Django foi adotado como *framework web* principal por oferecer, em uma mesma base, recursos de roteamento, modelos de dados, formulários, autenticação, administração, renderização de páginas e testes automatizados [Django Software Foundation 2025; Django Software Foundation 2025; Django Software Foundation 2025]. Essa integração é adequada para sistemas de gestão acadêmica porque reduz a necessidade de combinar muitos compo-

nentes externos para resolver tarefas essenciais, como persistir entidades, validar formulários, controlar acesso e organizar páginas.

No GHE, a divisão em aplicações Django independentes aproxima o código da divisão do domínio: instituição, base acadêmica, horários, indisponibilidade, consulta pública, secretaria, exportações e notificações. Essa estrutura facilita a evolução incremental do trabalho, pois permite alterar a geração de horários ou a coleta de indisponibilidade sem misturar essas responsabilidades com as páginas públicas.

2.3.2 PostgreSQL, Supabase e integridade relacional

O PostgreSQL foi utilizado como banco de dados relacional por sua capacidade de representar entidades fortemente relacionadas e aplicar restrições de integridade [The PostgreSQL Global Development Group 2025]. Essa escolha é relevante porque o problema de horários depende de vínculos entre campus, curso, período, disciplina, professor, sala, laboratório, semestre e faixa de horário. Sem integridade referencial, seria mais difícil garantir que a grade publicada e os rascunhos de geração permanecessem coerentes.

O Supabase foi empregado como serviço gerenciado de Postgres, oferecendo infraestrutura de banco sem exigir administração manual do servidor no recorte do projeto [Supabase 2025; Supabase 2025]. A aplicação Django se comunica com esse banco por meio do Psycopg, adaptador PostgreSQL para Python usado na configuração da persistência [Psycopg Team 2025; Psycopg Team 2025]. Essa combinação permitiu manter o modelo relacional como centro da solução e, ao mesmo tempo, preparar o sistema para implantação fora do ambiente local.

2.3.3 Django Templates, HTML, CSS e JavaScript

A interface do GHE foi desenvolvida com *Django Templates*, HTML, CSS e JavaScript pontual. A opção por páginas renderizadas no servidor, em vez de uma aplicação de página única, é compatível com o escopo do trabalho porque as telas dependem de dados relacionais, formulários e tabelas de consulta, não de interação em tempo real complexa [Django Software Foundation 2025]. O uso de templates também facilita a reutilização de componentes visuais e a manutenção de páginas públicas e restritas dentro do próprio projeto Django.

O HTML organiza a estrutura das páginas, o CSS define a apresentação e a responsividade, e o JavaScript complementa interações específicas, como filtros, movimentação visual de aulas e atualização de elementos da interface. Essa separação ajuda a manter a aplicação compreensível e suficiente para os fluxos implementados: consulta pública, matriz curricular, indisponibilidade docente, painel da secretaria, edição de rascunhos e edição da grade publicada.

2.3.4 Requests e Beautiful Soup na coleta de dados

Para aproximar o sistema da realidade do campus, foi necessário transformar a grade pública existente em uma base relacional. A biblioteca *Requests* foi usada para acessar as páginas públicas do portal de horários por HTTP, enquanto *Beautiful Soup* foi usada para interpretar o HTML e extrair cursos, disciplinas, professores, salas, laboratórios, dias e faixas de horário [Requests contributors 2025; Richardson 2025]. O resultado dessa etapa é um arquivo CSV que alimenta a importação por estágios.

Essa camada de coleta não foi concebida como fonte permanente de verdade do sistema. Ela funcionou como mecanismo de validação inicial e carga da base, permitindo testar o GHE com dados reais do IFPB, Campus Campina Grande. Depois da importação, o banco do GHE passa a armazenar os dados necessários para consulta, validação, geração e publicação.

2.3.5 Configuração de ambiente e implantação

As configurações de ambiente foram centralizadas com *django-environ*, biblioteca usada para separar segredos, endereços de conexão e parâmetros de implantação do código-fonte [django-environ contributors 2025]. Essa separação é importante porque informações como chave secreta, URL do banco, modo de depuração, domínios permitidos e configuração de e-mail não devem ser fixadas diretamente no código da aplicação.

No contexto do GHE, essa organização também permite diferenciar ambiente local, ambiente de testes e ambiente de produção. Em desenvolvimento, o sistema pode usar configurações mais simples para facilitar testes manuais. Em produção, parâmetros como `DEBUG=False`, `ALLOWED_HOSTS`, conexão SSL com o banco e cookies seguros passam a compor a configuração esperada de implantação.

2.3.6 Segurança web como requisito transversal

A segurança web foi tratada como requisito transversal da arquitetura, especialmente porque o GHE combina consulta pública, área restrita da secretaria e acesso individual de professores por identificador. A documentação do Django destaca recursos de proteção contra XSS, CSRF, injeção de SQL, validação de *Host*, uso de HTTPS e cookies seguros [Django Software Foundation 2026]. Além disso, o OWASP Top 10:2025 reúne riscos críticos para aplicações *web*, como controle de acesso quebrado, configuração insegura, falhas criptográficas, injeção e falhas de autenticação [OWASP Foundation 2025].

No escopo deste trabalho, essas referências servem como base para justificar escolhas como uso do ORM do Django, formulários com validação, proteção CSRF em submissões, separação de segredos por variáveis de ambiente, conexão segura com o banco e restrição da área operacional da secretaria. A segurança completa de um sistema em produção exigiria ainda monitoramento, política de senhas, revisão de permissões, proteção contra força bruta, auditoria de logs e rotina de atualização de dependências.

A Tabela 2.1 resume as tecnologias adotadas e o papel de cada uma no sistema, separando a infraestrutura principal, a camada de páginas, o banco de dados, a configuração de ambiente, o processo de coleta dos dados externos e os mecanismos de segurança utilizados.

Tecnologia	Papel no sistema
Django	<i>Framework web</i> principal, responsável por modelos de dados, visões, endereços, administração, autenticação, formulários e suíte de testes.
<i>Django Templates</i> , HTML, CSS e JavaScript	Renderização das páginas públicas, da interface de indisponibilidade e da área da secretaria sem uso de aplicação de página única.
PostgreSQL	Banco de dados relacional com integridade referencial, restrições de integridade e consultas estruturadas.
Supabase	Serviço gerenciado de Postgres utilizado como infraestrutura de banco do projeto.
Psycopg	Adaptador Python-PostgreSQL empregado pela aplicação Django.
<i>django-environ</i>	Leitura de variáveis de ambiente e centralização da configuração do projeto.
<i>Requests</i>	Coleta HTTP das páginas públicas de horários do IFPB.
<i>Beautiful Soup</i>	Extração e interpretação do conteúdo HTML utilizado no processo de coleta da grade.
Django Security Middleware e configurações de produção	Apoio à proteção contra CSRF, XSS, validação de <i>Host</i> , HTTPS e cookies seguros no ambiente implantado.

Tabela 2.1: *Tecnologias utilizadas no GHE e seu papel no projeto.*

Essa combinação de tecnologias atende ao estágio atual do trabalho por oferecer rapidez de desenvolvimento, clareza arquitetural e base suficiente para operação real. A geração automática foi implementada dentro do próprio domínio Django, aproveitando modelos de dados, restrições de integridade, transações e testes automatizados para manter rastreabilidade entre rascunho, validação e publicação.

Capítulo 3

Desenvolvimento do GHE

3.1 Requisitos e regras de negócio

O desenvolvimento do GHE foi guiado por requisitos funcionais, requisitos não funcionais, regras de negócio e fases de evolução registradas no repositório do projeto. Para tornar esse escopo mais explícito no texto da monografia, a Tabela 3.1 resume os requisitos funcionais considerados no recorte final do trabalho, indicando o ator principal e a situação de atendimento.

Tabela 3.1: *Requisitos funcionais do recorte final do GHE.*

Código	Descrição	Ator principal	Situação
RF01	Permitir consulta pública de horários sem autenticação de alunos.	Comunidade acadêmica	Implementado
RF02	Organizar a navegação por instituição, campus, modalidade, curso e período.	Comunidade acadêmica	Implementado
RF03	Permitir visualização de horários por professor, sala e laboratório.	Comunidade acadêmica	Implementado
RF04	Exibir a matriz curricular por curso e período, evidenciando disciplinas vinculadas à oferta.	Coordenação e comunidade	Implementado
RF05	Coletar e importar a grade pública do IFPB, Campus Campina Grande, para uma base relacional.	Secretaria	Implementado
RF06	Permitir preenchimento privado de indisponibilidade docente por identificador individual.	Professor	Implementado
RF07	Permitir gestão de vínculos professor-disciplina por semestre, curso, período e turma.	Secretaria	Implementado
RF08	Permitir configuração de horários liberados para geração por curso e semestre.	Secretaria	Implementado
RF09	Gerar rascunhos de grade a partir de vínculos, horários permitidos e indisponibilidades.	Secretaria	Implementado
RF10	Validar rascunhos contra conflitos de professor, turma, sala, laboratório, indisponibilidade, horários permitidos e carga semanal.	Secretaria	Implementado
RF11	Publicar rascunhos validados como grade oficial do curso e semestre.	Secretaria	Implementado
RF12	Permitir edição posterior dos horários publicados pela secretaria.	Secretaria	Implementado

Código	Descrição	Ator principal	Situação
RF13	Exportar horários em PDF e planilha.	Secretaria e comunidade	Planejado
RF14	Permitir boletim de notificações por período para alterações na grade.	Comunidade acadêmica	Planejado

As regras de negócio mais relevantes determinam que o sistema seja público para consulta, sem autenticação de alunos, e que a indisponibilidade docente represente horários nos quais o professor não pode ministrar aulas. Também estabelecem que cada curso, professor, sala, laboratório e horário pertença a um campus específico, e que o sistema preserve coerência entre essas entidades para evitar conflitos e inconsistências.

No motor de geração, as regras duras foram tratadas como condições que impedem ou invalidam uma alocação: professor em duas aulas no mesmo horário, período ou turma com aulas simultâneas, sala ocupada por mais de uma aula, laboratório ocupado por mais de uma aula, uso de horário não liberado para o curso e conflito com indisponibilidade docente. As regras suaves aparecem como critérios de qualidade da alocação, como preferência por blocos compactos e melhor aproveitamento da semana, mas sem se sobrepor às restrições obrigatórias.

Com base nisso, o desenvolvimento foi organizado em fases: base do projeto, cadastro acadêmico, consulta pública, indisponibilidade docente, geração de horários, publicação da grade, edição pela secretaria, exportações e extras. Esta monografia cobre a implementação efetiva até o ponto em que a base do sistema, a consulta pública, a coleta de indisponibilidade, a geração automática e a publicação da grade já estão operacionais.

3.2 Casos de uso do sistema

Os casos de uso do GHE foram organizados a partir dos atores que interagem diretamente com o sistema: comunidade acadêmica, professores e secretaria. A Figura 3.1 apresenta uma visão geral desses atores e das principais funcionalidades disponíveis no recorte implementado.

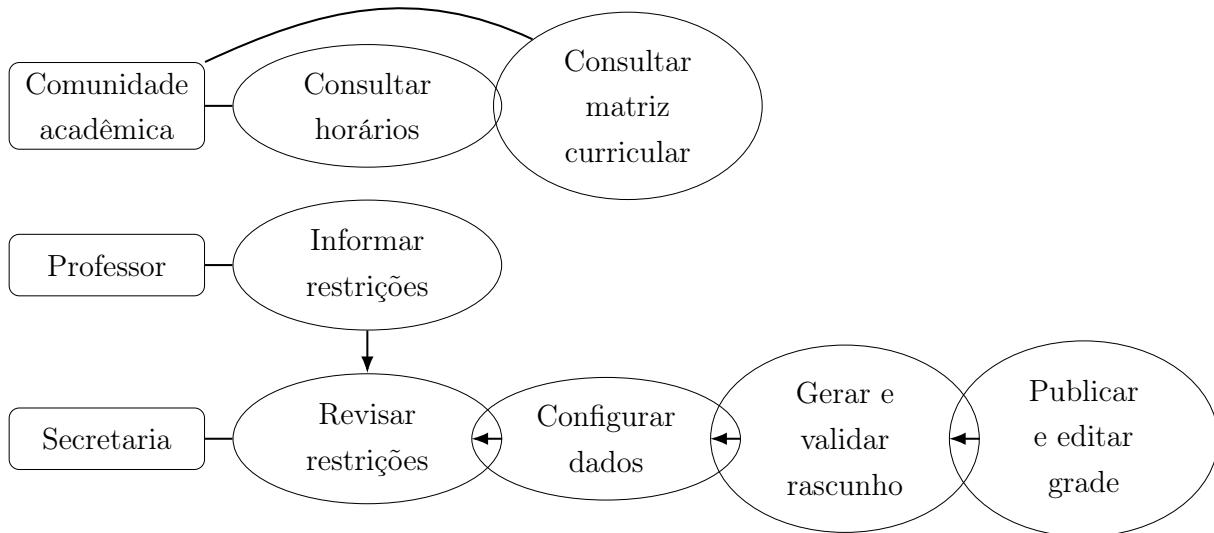


Figura 3.1: Diagrama geral de casos de uso do GHE.

No fluxo operacional da secretaria, os casos de uso formam uma sequência controlada: preparar os dados de entrada, gerar um rascunho, validar conflitos, publicar a grade e realizar ajustes posteriores. A Figura 3.2 detalha esse fluxo, destacando que a publicação não ocorre automaticamente sem validação.

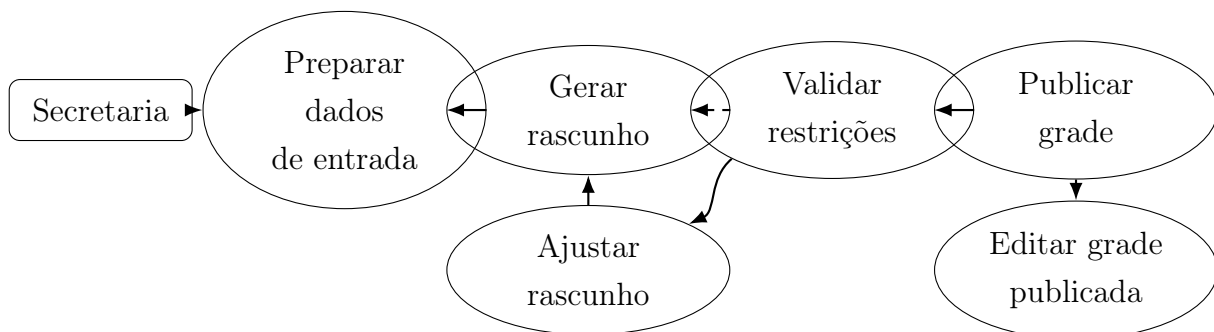


Figura 3.2: Casos de uso do fluxo de geração, validação e publicação pela secretaria.

3.3 Arquitetura modular do sistema

Para separar responsabilidades e facilitar a evolução do código, o GHE foi dividido em aplicações Django independentes. Essa decisão ajuda a manter o domínio do problema mais organizado e reduz o acoplamento entre funcionalidades públicas, acadêmicas e operacionais. A Tabela 3.2 sintetiza essa divisão, indicando o papel principal de cada aplicação no conjunto do sistema.

Além da divisão por aplicações, a implementação procurou manter a regra de negócio fora das visões sempre que possível. Serviços específicos concentram montagem de contexto, coleta automatizada, importação por etapas, construção da matriz curricular, fluxo de indisponibilidade, geração de rascunho, validação e publicação. Essa organização torna o código mais legível e favorece testes direcionados.

Aplicação	Responsabilidade principal
<i>institutions</i>	Cadastro de instituição e campus, base da segmentação multi-instituição e multi-campus.
<i>academic</i>	Modalidades, cursos, períodos, semestres letivos, professores, disciplinas, salas e laboratórios.
<i>scheduling</i>	Modelagem dos horários, vínculos de ensino, horários permitidos, rascunhos, validação e publicação.
<i>availability</i>	Coleta restrita de indisponibilidade docente por campanha e identificador individual de acesso.
<i>public</i>	Páginas públicas, navegação por filtros, coleta automatizada, importação da grade externa e prévia de rascunho.
<i>secretariat</i>	Área restrita da secretaria para operar geração, revisão, publicação e edição de horários.
<i>exports</i>	Estrutura inicial para futuras exportações em PDF e planilha.
<i>notifications</i>	Estrutura inicial para comunicação e notificações por período.

Tabela 3.2: *Síntese das aplicações do GHE e suas responsabilidades.*

3.4 Modelo de dados principal

3.4.1 Modelagem do banco de dados

A modelagem do banco de dados do GHE foi organizada em torno de um modelo relacional, persistido em PostgreSQL e acessado pela aplicação por meio dos modelos do Django. Essa escolha permite representar explicitamente as dependências entre campus, cursos, períodos, disciplinas, professores, espaços físicos, restrições de disponibilidade e grades geradas. A separação em entidades especializadas reduz duplicidade, facilita validações por chaves estrangeiras e mantém a base adequada tanto para consulta pública quanto para operação da secretaria.

O modelo parte das entidades *Institution* e *Campus*, responsáveis por definir o escopo institucional e físico do sistema. A partir desse núcleo, o módulo acadêmico associa cursos, professores, salas, laboratórios e termos letivos sempre a um campus, evitando que dados de unidades diferentes sejam misturados. A Figura 3.3 exemplifica o recorte principal da modelagem, destacando como a base acadêmica alimenta as restrições de entrada, o rascunho de geração e a grade publicada.

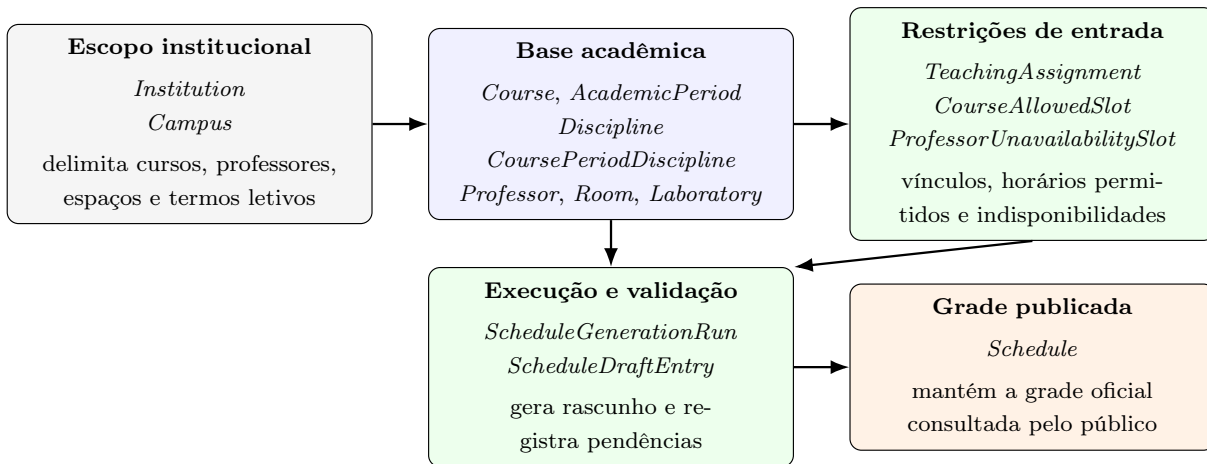


Figura 3.3: Exemplo simplificado da modelagem relacional usada para estruturar a base acadêmica, as restrições de entrada, os rascunhos e a grade publicada.

Na camada acadêmica, *Course* incorpora o campo *course_shift*, permitindo diferenciar cursos com o mesmo nome em turnos distintos. *AcademicTerm* representa o semestre letivo do campus e identifica qual semestre deve ser considerado atual na consulta pública. *AcademicPeriod* representa os períodos do curso, enquanto *Discipline* armazena as disciplinas vinculadas ao curso. Para tornar explícita a matriz curricular, foi criada a entidade *CoursePeriodDiscipline*, que relaciona curso, período e disciplina.

No módulo de horários, a entidade *Schedule* centraliza a grade publicada. Ela vincula campus, curso, período, semestre letivo, professor, disciplina, dia da semana, faixa de horário e espaço físico. O modelo também inclui o campo *class_group* (grupo ou turma da aula), além da possibilidade de registrar simultaneamente sala e laboratório em uma mesma aula, comportamento necessário para refletir a grade real importada do campus.

Para a geração automática, o sistema utiliza *TeachingAssignment*, *CourseAllowedSlot*, *ScheduleGenerationRun* e *ScheduleDraftEntry*. O vínculo professor-disciplina define quais aulas precisam ser geradas em determinado semestre. Os horários permitidos delimitam as faixas que o algoritmo pode usar por curso. Cada execução de geração registra sua situação operacional e seu resumo, e as entradas de rascunho armazenam a grade gerada antes da publicação final.

No módulo de disponibilidade, *AvailabilityCampaign* representa uma campanha de coleta por campus e período letivo. *ProfessorAvailability* identifica a submissão individual de cada professor por meio de um identificador único de acesso, e *ProfessorUnavailabilitySlot* registra as faixas semanais marcadas como indisponíveis. Essas informações são reutilizadas pelo gerador para evitar alocações em horários bloqueados pelo docente.

3.4.2 Modelo relacional do banco de dados

O modelo relacional do banco de dados detalha como as entidades descritas anteriormente são materializadas em tabelas, atributos e chaves estrangeiras. Cada tabela possui uma

chave primária interna, representada por *id*, e os vínculos entre tabelas são implementados por campos de referência, como *campus_id*, *course_id*, *academic_term_id*, *professor_id* e *discipline_id*. Essa estrutura permite que o sistema mantenha a integridade do escopo acadêmico, evitando, por exemplo, que uma disciplina de um curso seja associada a um período de outro curso ou que um professor de campus diferente seja usado em uma grade local.

A Figura 3.4 apresenta um exemplo do modelo relacional do GHE, com as principais entidades do recorte implementado, seus atributos mais relevantes e as relações entre elas. As cardinalidades indicam que um registro de origem pode estar associado a vários registros dependentes, enquanto as setas tracejadas representam relações operacionais usadas pelo processo de geração e publicação da grade.

No modelo apresentado, *CoursePeriodDiscipline* funciona como tabela associativa da matriz curricular, ligando curso, período e disciplina. *TeachingAssignment* relaciona professor, disciplina, período, curso e semestre letivo para definir a carga que deve ser alocada. *CourseAllowedSlot* restringe os horários possíveis por curso e semestre, enquanto *ProfessorUnavailabilitySlot* bloqueia faixas informadas pelos docentes. Ao final do processamento, *ScheduleGenerationRun* registra a execução, *ScheduleDraftEntry* armazena o rascunho validado e *Schedule* concentra a grade publicada.

Além das chaves estrangeiras, o banco utiliza restrições de unicidade para evitar duplicidade em pontos críticos, como curso por campus e turno, período por curso, disciplina por curso, vínculo curso-período-disciplina, horário permitido por curso e semestre, submissão de disponibilidade por professor e campanha, item de rascunho por execução e item publicado por turma, disciplina, professor, dia e horário. Essas restrições complementam as validações da aplicação e tornam o próprio banco uma camada de proteção contra inconsistências.

Essa separação também diferencia dados relativamente estáveis, como campus, cursos, professores e espaços físicos, de dados operacionais, como campanhas de disponibilidade, vínculos de ensino, execuções de geração e rascunhos. Com isso, a importação pode atualizar a base acadêmica sem recriar toda a grade, a consulta pública consegue filtrar horários por curso, professor, sala ou laboratório, e o motor de geração reutiliza as mesmas referências para validar escopo, disponibilidade e conflitos antes da publicação.

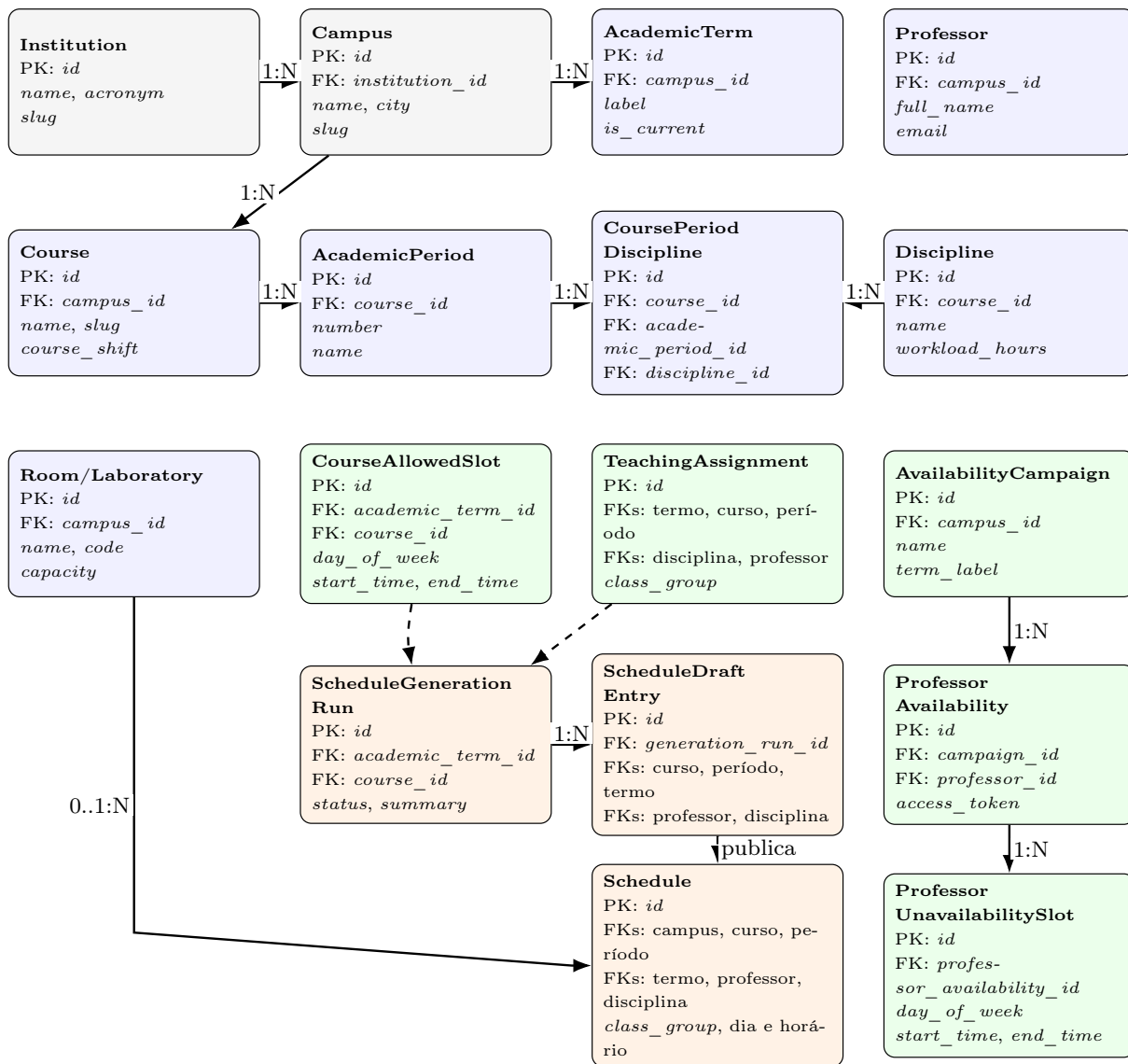


Figura 3.4: Exemplo de modelo relacional do banco de dados do GHE com entidades, atributos principais, cardinalidades e relações operacionais de geração.

3.5 Segurança de dados e segurança web

A modelagem do GHE diferencia dados públicos, dados operacionais restritos e dados de configuração. Os horários publicados, informações de curso, matriz curricular, professores, salas e laboratórios compõem a camada pública de consulta. Já vínculos professor-disciplina, campanhas de indisponibilidade, rascunhos de geração, ações de publicação e configuração de semestre atual pertencem à operação da secretaria e exigem acesso restrito. As indisponibilidades docentes, embora usadas apenas como restrições de geração, também devem ser tratadas com cuidado por representarem preferências e limitações individuais de trabalho.

No acesso dos professores, a versão implementada utiliza identificador individual de acesso para cada submissão de indisponibilidade. Essa solução reduz a complexidade da versão mínima viável, pois dispensa cadastro completo de professores, mas exige proteção do enlace,

uso de identificadores não previsíveis e envio controlado pela secretaria. Para trabalhos futuros, a autenticação plena dos docentes pode substituir ou complementar esse fluxo, com controle de senha, expiração de acesso, histórico de alterações e trilha de auditoria mais detalhada.

Na área da secretaria, o acesso é restrito por autenticação e permissões do sistema. As ações que alteram estado, como geração, publicação, edição de horários e definição de semestre atual, devem permanecer associadas a usuários autorizados. Essa separação reduz o risco de alteração indevida da grade pública e atende ao princípio de controle de acesso, apontado pelo OWASP Top 10:2025 como um dos principais riscos em aplicações *web* [OWASP Foundation 2025].

Do ponto de vista da segurança de aplicação, o uso de Django contribui com proteções importantes. O mecanismo de *templates* aplica escape automático para reduzir riscos de XSS, formulários submetidos por POST usam proteção CSRF, e consultas feitas pelo ORM utilizam parametrização para reduzir risco de injeção de SQL [Django Software Foundation 2026]. Mesmo assim, essas proteções não eliminam a necessidade de validação de entradas, cuidado com marcação HTML considerada segura e revisão de permissões em cada tela restrita.

Na implantação, a segurança também depende de configuração adequada. O projeto prevê separação de segredos por variáveis de ambiente, uso de `DEBUG=False` em produção, definição explícita de `ALLOWED_HOSTS`, conexão com banco por SSL, redirecionamento para HTTPS quando disponível, cookies seguros para sessão e CSRF, e proteção da chave secreta da aplicação. Esses pontos são relevantes porque falhas de configuração e falhas criptográficas também aparecem entre os riscos recorrentes de aplicações *web* [OWASP Foundation 2025].

Assim, a segurança do GHE deve ser entendida como combinação de modelagem, controle de acesso, configuração de implantação e práticas de desenvolvimento. No recorte deste trabalho, foram priorizadas autenticação da secretaria, isolamento entre consulta pública e operação restrita, identificadores individuais para professores, validação de formulários, uso do ORM e configuração segura de banco. Como limitações atuais, permanecem a necessidade de autenticação plena dos professores, política de expiração de identificadores, auditoria mais detalhada de alterações e proteção adicional contra tentativas repetidas de login.

3.6 Fluxo de coleta e importação dos dados

Como o objetivo do sistema era trabalhar com dados reais do IFPB, foi implementado um fluxo de coleta e importação específico para capturar e normalizar a grade publicada no portal institucional de horários. A coleta é feita com *Requests* [Requests contributors 2025], responsável pelas requisições HTTP, e *Beautiful Soup* [Richardson 2025], responsável por navegar e extrair as tabelas e metadados do HTML.

A coleta automatizada produz um arquivo CSV contendo informações sobre curso, mo-

dalidade, turno, período, disciplina, professor, sala, laboratório, turma, dia da semana, faixa de horário, semestre e endereço de origem. Na sequência, o projeto executa uma importação em etapas e idempotente, capaz de criar ou atualizar as entidades do banco em ordem de dependência sem duplicar registros já existentes: instituição, campus, modalidades, cursos, períodos, professores, salas, laboratórios, disciplinas, vínculos curso-período-disciplina, vínculos professor-disciplina e, por fim, horários publicados.

Esse fluxo foi projetado para lidar com dados incompletos ou ambíguos. Valores não resolvidos, como marcadores provisórios (*placeholders*) e inconsistências de nomenclatura, são separados para revisão manual sem interromper toda a carga. Na prática, isso permite transformar uma grade pública originalmente voltada à visualização em uma base estruturada para consulta, análise, geração e publicação controlada.

3.7 Módulos funcionais implementados

3.7.1 Consulta pública de horários

A consulta pública do GHE foi organizada para permitir navegação por diferentes recortes. A página central de horários funciona como núcleo inicial de navegação, permitindo que o usuário escolha entre cursos, laboratórios, salas e professores. No caso dos cursos, a navegação ocorre por modalidade e depois por curso e período, o que torna a exploração da grade mais próxima da organização acadêmica adotada pelo campus. A Figura 3.5 apresenta essa central de navegação.



Figura 3.5: Central de navegação de horários do GHE, organizada por modalidade, curso e período.

Além do núcleo inicial, o sistema oferece páginas dedicadas para curso, professor, sala e laboratório. Essas páginas mantêm um filtro principal pré-definido e apresentam a grade semanal em formato matricial, com identificação de disciplina, turma, curso, período, professor e espaço associado. A Figura 3.6 mostra a grade semanal publicada do curso de Telemática.

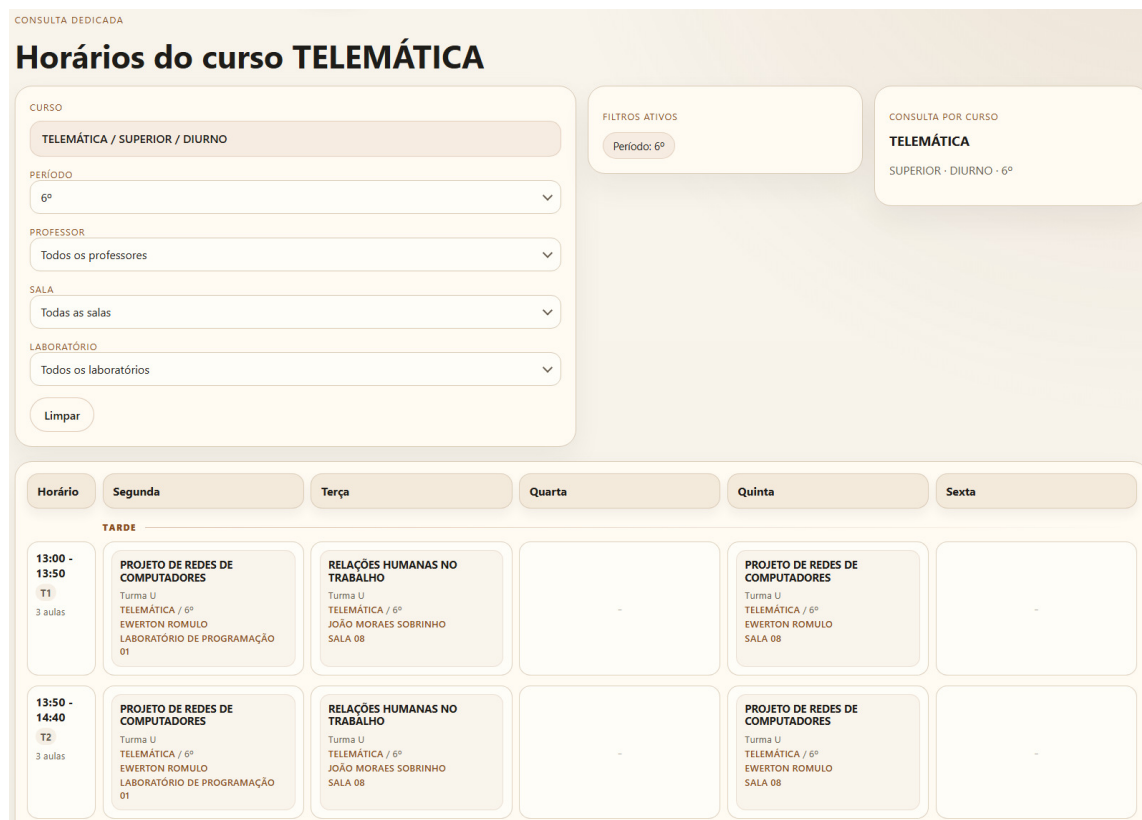


Figura 3.6: Grade semanal pública do curso de Telemática no GHE.

Para facilitar consultas específicas, a página dedicada de professor também permite refinamento adicional por cursos, salas e laboratórios relacionados àquele docente. Isso favorece tanto a visualização pública quanto a auditoria da grade, como exemplifica a Figura 3.7.

3.7.2 Matriz curricular

Outro módulo implementado foi a visualização da matriz curricular por curso e período. Essa funcionalidade utiliza explicitamente a relação entre *Course*, *AcademicPeriod* e *CoursePeriodDiscipline*, permitindo ao sistema distinguir a estrutura curricular da grade semanal em si. A Figura 3.8 apresenta a matriz pública do curso de Telemática.

3.7.3 Coleta de indisponibilidade docente

No módulo de disponibilidade, foi implementado um fluxo restrito baseado em identificador individual de acesso. Cada campanha gera submissões por professor, e o docente acessa um endereço exclusivo para marcar as faixas de horário em que não pode ministrar aulas. O

CONSULTA DEDICADA

Horários de DANIELLA DIAS CAVALCANTE DA SILVA

CURSO: TELEMÁTICA - DIURNO

PERÍODO: 6º

PROFESSOR: DANIELLA DIAS CAVALCANTE DA SILVA

SALA: Todas as salas

LABORATÓRIO: Todos os laboratórios

Limpar

FILTROS ATIVOS: Curso: TELEMÁTICA Período: 6º

CONSULTA POR PROFESSOR: DANIELLA DIAS CAVALCANTE DA SILVA Currículo Lattes

Horário	Segunda	Terça	Quarta	Quinta	Sexta
TARDE					
14:40 - 15:30 T3 1 aula	-	-	-	TRABALHO DE CONCLUSÃO DE CURSO Turma A TELEMÁTICA / 6º DANIELLA DIAS CAVALCANTE DA SILVA SALA 08	-
15:50 - 16:40 T4 1 aula	-	-	-	TRABALHO DE CONCLUSÃO DE CURSO Turma A TELEMÁTICA / 6º DANIELLA DIAS CAVALCANTE DA SILVA SALA 08	-

Figura 3.7: Página dedicada de professor com refinamento contextual por cursos, salas e laboratórios.

formulário é construído em grade semanal, reutilizando as faixas já existentes no campus, o que aproxima a coleta da realidade operacional da geração de horários.

Além do formulário individual, o projeto oferece criação em lote de submissões, exportação de endereços individuais, exportação da situação da campanha, exportação detalhada das faixas de indisponibilidade e envio de convites por e-mail com serviço configurável. A secretaria também pode revisar e editar manualmente a indisponibilidade de um professor pela área restrita, como mostra a Figura 3.9.

3.7.4 Área da secretaria

A área da secretaria centraliza o fluxo operacional da geração de horários. Ela permite selecionar semestre e curso, acompanhar vínculos ativos, horários liberados, aulas no rascunho, aulas publicadas, situação da validação e informações da última publicação. A Figura 3.10 mostra o painel principal após a geração e publicação de um rascunho.



Figura 3.8: Matriz curricular pública do curso de Telemática no GHE.



Figura 3.9: Edição de indisponibilidade docente pela área da secretaria.



Figura 3.10: Painel da secretaria com indicadores de geração, validação e publicação.

Na etapa inicial do fluxo, a secretaria revisa os vínculos professor-disciplina do semestre. Cada vínculo informa curso, período, disciplina, professor, turma, quantidade semanal de aulas, preferências de sala ou laboratório e a possibilidade de quatro aulas consecutivas quando aplicável. A tela também oferece cópia de vínculos entre semestres, evitando recadastro manual quando uma oferta é reaproveitada. A Figura 3.11 apresenta essa tela de configuração.



Figura 3.11: Tela de vínculos professor-disciplina usada como entrada do gerador.

Outra entrada do gerador é a matriz de horários permitidos. A secretaria marca, por curso e semestre, quais faixas de segunda a sexta podem ser utilizadas. Essa configuração impede que o algoritmo aloque aulas em turnos inadequados ou em horários bloqueados institucionalmente. A Figura 3.12 mostra essa configuração operacional.



Figura 3.12: Configuração de horários permitidos por curso e semestre.

3.7.5 Motor de geração, validação e publicação

A geração automática do GHE foi implementada como um serviço determinístico. Ao iniciar uma execução, o sistema carrega os vínculos professor-disciplina ativos do curso e semestre, os horários permitidos, o conjunto de salas e laboratórios do campus e as indisponibilidades docentes registradas. Em seguida, ordena as disciplinas por período, carga semanal e identificação da disciplina, formando blocos de duas aulas consecutivas e, quando permitido no vínculo, blocos de quatro aulas consecutivas.

Durante a alocação, o serviço evita conflitos de professor, período ou turma, sala, laboratório e indisponibilidade docente. Também considera rascunhos de outros cursos do mesmo semestre para evitar ocupação simultânea de professores e recursos compartilhados. Quando não consegue alocar toda a carga de uma disciplina, a execução é concluída com pendências e registra um resumo explicativo para revisão.

O resultado da geração é armazenado como *ScheduleDraftEntry*, sem alterar imediatamente a grade pública. O rascunho pode ser visualizado em uma matriz semanal e ajustado por arrastar e soltar para outros horários liberados, como mostra a Figura 3.13.

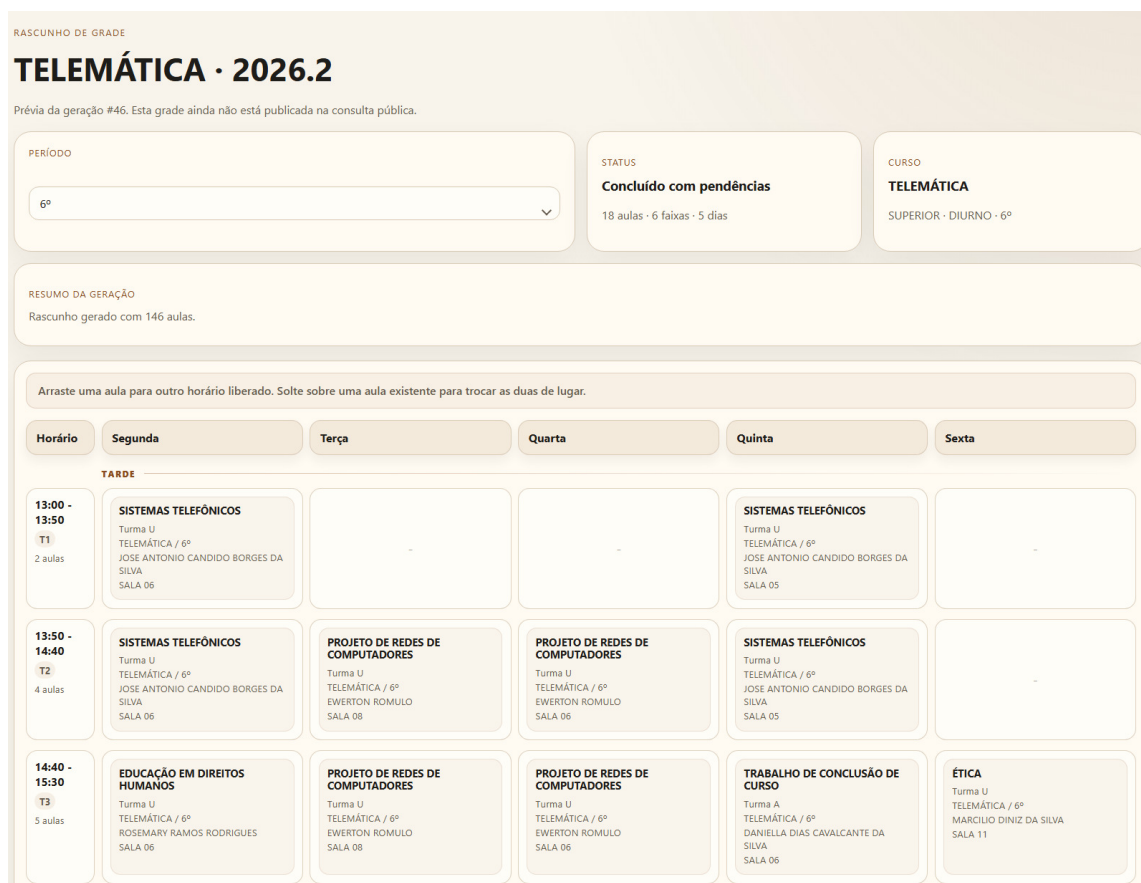


Figura 3.13: Prévia do rascunho gerado antes da publicação na consulta pública.

Antes da publicação, o rascunho passa por validação. O relatório verifica conflitos de professor, turma ou período, sala, laboratório, aulas fora dos horários permitidos, indisponibilidades docentes, grupos não pareados, quatro aulas consecutivas sem permissão e cargas

incompletas. Quando a validação não encontra problemas, a publicação substitui a grade pública daquele curso e semestre pelos itens do rascunho. A Figura 3.14 exemplifica o retorno de validação sem pendências.

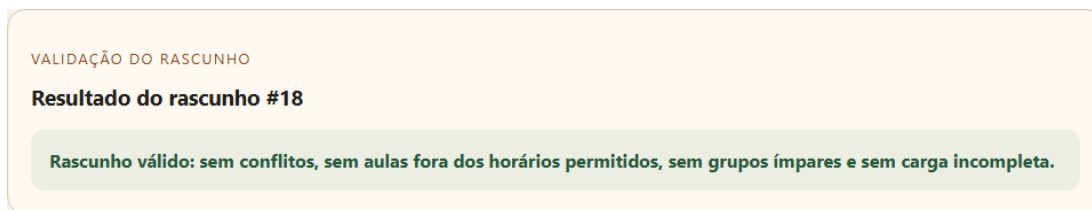


Figura 3.14: *Validação do rascunho antes da publicação da grade.*

Após a publicação, a secretaria ainda pode editar a grade oficial. A tela de edição utiliza a mesma visualização semanal do rascunho e permite mover aulas para outros horários liberados, trocar aulas entre si e ajustar professor, sala, laboratório, turma e observações por disciplina. A Figura 3.15 ilustra essa edição posterior à publicação.

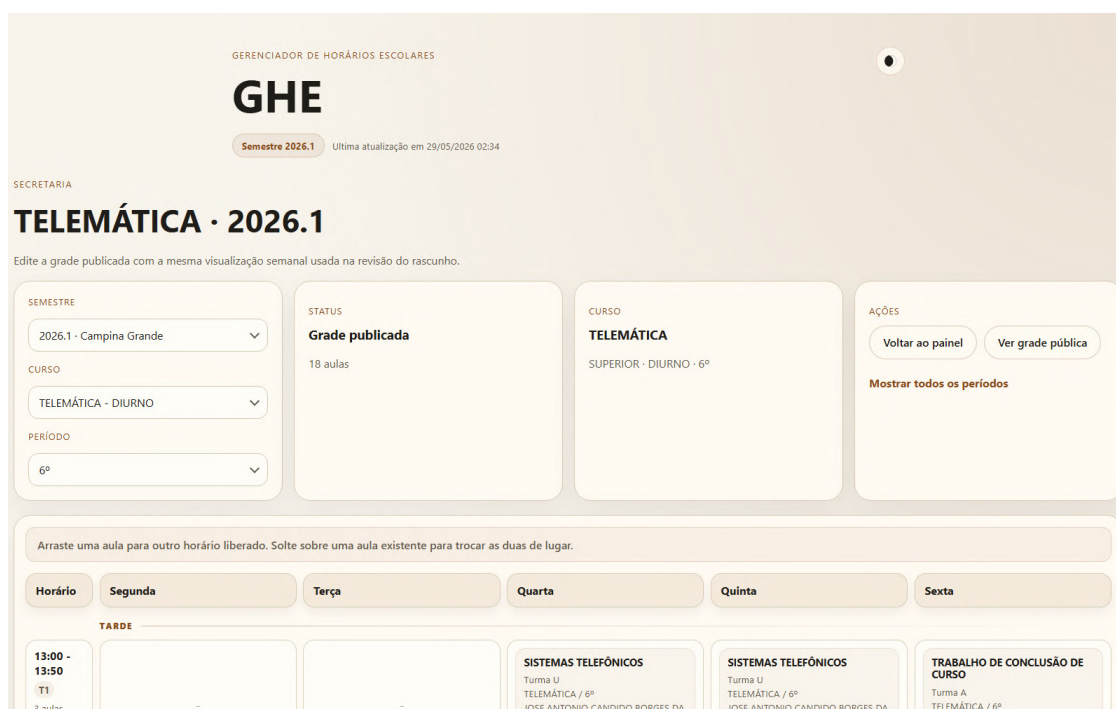


Figura 3.15: *Edição da grade publicada pela secretaria após a publicação do rascunho.*

Além da edição direta da grade publicada, o resultado da geração pode ser conferido na própria matriz semanal, já com as aulas alocadas nos dias e horários correspondentes. A Figura 3.16 mostra o horário gerado para o 6º período de Telemática após a execução da geração.

Horário	Segunda	Terça	Quarta	Quinta	Sexta
TARDE					
13:00 - 13:50 T1 3 aulas	PROJETO DE REDES DE COMPUTADORES Turma U TELEMÁTICA / 6º EWERTON ROMULO LABORATÓRIO DE PROGRAMAÇÃO 01	RELAÇÕES HUMANAS NO TRABALHO Turma U TELEMÁTICA / 6º JOÃO MORAES SOBRINHO SALA 08	-	PROJETO DE REDES DE COMPUTADORES Turma U TELEMÁTICA / 6º EWERTON ROMULO SALA 08	-
13:50 - 14:40 T2 3 aulas	PROJETO DE REDES DE COMPUTADORES Turma U TELEMÁTICA / 6º EWERTON ROMULO LABORATÓRIO DE PROGRAMAÇÃO 01	RELAÇÕES HUMANAS NO TRABALHO Turma U TELEMÁTICA / 6º JOÃO MORAES SOBRINHO SALA 08	-	PROJETO DE REDES DE COMPUTADORES Turma U TELEMÁTICA / 6º EWERTON ROMULO SALA 08	-
14:40 - 15:30 T3 3 aulas	ÉTICA Turma U TELEMÁTICA / 6º MARCILIO DINIZ DA SILVA SALA 08	SISTEMAS TELEFÔNICOS Turma U TELEMÁTICA / 6º JOSE ANTONIO CANDIDO BORGES DA SILVA SALA 08	-	TRABALHO DE CONCLUSÃO DE CURSO Turma A TELEMÁTICA / 6º DANIELLA DIAS CAVALCANTE DA SILVA SALA 08	-
15:50 - 16:40 T4 3 aulas	ÉTICA Turma U TELEMÁTICA / 6º MARCILIO DINIZ DA SILVA SALA 08	SISTEMAS TELEFÔNICOS Turma U TELEMÁTICA / 6º JOSE ANTONIO CANDIDO BORGES DA SILVA SALA 08	-	TRABALHO DE CONCLUSÃO DE CURSO Turma A TELEMÁTICA / 6º DANIELLA DIAS CAVALCANTE DA SILVA SALA 08	-
16:40 - 17:30 T5 3 aulas	EDUCAÇÃO EM DIREITOS HUMANOS Turma U TELEMÁTICA / 6º ROSEMARY RAMOS RODRIGUES SALA 08	SISTEMAS TELEFÔNICOS Turma U TELEMÁTICA / 6º JOSE ANTONIO CANDIDO BORGES DA SILVA SALA 08	-	TRABALHO DE CONCLUSÃO DE CURSO Turma A TELEMÁTICA / 6º DANIELLA DIAS CAVALCANTE DA SILVA SALA 08	-
17:30 - 18:20 T6 3 aulas	EDUCAÇÃO EM DIREITOS HUMANOS Turma U TELEMÁTICA / 6º ROSEMARY RAMOS RODRIGUES SALA 08	SISTEMAS TELEFÔNICOS Turma U TELEMÁTICA / 6º JOSE ANTONIO CANDIDO BORGES DA SILVA SALA 08	-	TRABALHO DE CONCLUSÃO DE CURSO Turma A TELEMÁTICA / 6º DANIELLA DIAS CAVALCANTE DA SILVA SALA 08	-

Figura 3.16: Horário gerado para o 6º período de Telemática após a execução da geração.

Capítulo 4

Resultados Obtidos

Este capítulo apresenta o estado atual validado do GHE, considerando as funcionalidades implementadas e verificadas no projeto. Os resultados obtidos permitem analisar a consolidação da base acadêmica, a operação das interfaces públicas, a coleta de indisponibilidade docente e o fluxo completo de geração, validação, publicação e edição da grade.

4.1 Base institucional e acadêmica importada

Após a execução do fluxo de coleta automatizada e importação por etapas, o GHE passou a manter uma base consistente do IFPB, Campus Campina Grande. O conjunto de dados atualmente carregado demonstra que o projeto opera com volume real suficiente para validar a modelagem proposta, as consultas públicas e a geração de horários. A Tabela 4.1 apresenta as principais métricas dessa base importada.

Elemento	Previsto	Realizado	Situação
Instituições	1	1	1/1 no recorte IFPB
Campi	1	1	1/1 para Campina Grande
Modalidades	4	4	4/4 importadas
Cursos	23	23	23/23 diferenciados por <i>course shift</i>
Períodos	95	95	95/95 vinculados aos cursos
Professores	174	174	174/174 importados para o campus
Disciplinas	783	783	783/783 associadas aos cursos
Vínculos curso-período-disciplina	793	793	793/793 na matriz curricular
Salas	35	35	35/35 vinculadas ao campus
Laboratórios	34	34	34/34 vinculados ao campus
Horários publicados	2156	2156	2156/2156 registros importados do portal público

Tabela 4.1: Métricas previstas e realizadas da base importada no GHE.

Na tabela, o valor previsto corresponde ao recorte necessário para validar a carga inicial do campus a partir dos dados coletados, e o valor realizado corresponde ao total efetivamente importado e disponível no banco. Esse resultado é importante porque confirma a viabilidade do modelo relacional adotado, especialmente no que diz respeito à separação por campus, ao vínculo explícito entre disciplina e período e à representação de aulas que utilizam simultaneamente sala e laboratório. Também demonstra que a importação por etapas foi capaz

de transformar a grade pública externa em uma base navegável, validável e reutilizável pelo gerador.

4.2 Consulta pública e organização da grade

No módulo público, os resultados mais relevantes estão na navegação por diferentes recortes de consulta e na visualização semanal da grade. O sistema permite acessar horários por curso, período, professor, sala e laboratório, além de centralizar a navegação inicial em um núcleo organizado por seção e modalidade.

Outro resultado importante é a matriz curricular pública, que passa a exibir a distribuição das disciplinas por período do curso selecionado. Em conjunto, essas telas demonstram que o sistema entrega uma base funcional para consulta pública mais estruturada que a simples publicação tabular da grade. Também indicam que a modelagem já permite expansões como exportação de horários, auditoria da grade e integração com outras fontes institucionais.

4.3 Coleta e gestão de indisponibilidade docente

No módulo de disponibilidade, o projeto disponibiliza campanhas de coleta, submissões individuais por professor e registros detalhados de indisponibilidade semanal. O acesso por identificador individual permite que o professor informe seus horários bloqueados sem exigir autenticação completa, preservando a simplicidade da versão mínima viável.

Além da coleta pelo professor, a área da secretaria permite consultar a matriz de indisponibilidades por campanha e editar manualmente as faixas de horário de um docente quando necessário. Esse resultado é relevante porque aproxima a coleta da operação real da coordenação: a secretaria pode acompanhar respostas, corrigir dados e garantir que o gerador utilize restrições atualizadas.

4.4 Geração automática de horários

O principal avanço em relação ao recorte inicial do projeto é a implementação da geração automática. O sistema já permite preparar os vínculos professor-disciplina do semestre, configurar horários permitidos, considerar indisponibilidades docentes, gerar rascunhos, validar pendências e publicar a grade oficial.

Em uma execução local com a base do curso de Telemática no semestre 2026.1, o gerador produziu um rascunho com 148 aulas. A validação do rascunho indicou ausência de pendências, permitindo a publicação da grade. Na publicação, 148 aulas foram criadas e 148 registros anteriores do mesmo curso e semestre foram substituídos. Esse comportamento confirma o fluxo previsto: a geração cria um rascunho isolado, a validação protege a publicação e a ação de publicar troca a grade pública apenas após aprovação. A Tabela 4.2 resume

os resultados observados nessa execução.

Etapas	Previsto	Realizado	Situação
Vínculos professor-disciplina	42 vínculos	42 vínculos	42/42 ativos no curso e semestre analisados
Horários permitidos	30 faixas	30 faixas	30/30 liberadas para uso pelo gerador
Rascunho gerado	148 aulas	148 aulas	148/148 aulas criadas no rascunho
Validação do rascunho	Sem pendências impeditivas	Sem pendências identificadas	Rascunho validado para publicação
Publicação	148 aulas do rascunho	148 criadas e 148 substituídas	148/148 publicadas no curso e semestre
Consulta pública	Grade do curso disponível	Grade publicada disponível	Atendido nas páginas públicas do curso

Tabela 4.2: Resultado de geração e publicação para o curso de Telemática no semestre 2026.1.

O relatório de validação cobre os principais riscos do problema: conflitos de professor, conflitos de turma ou período, conflitos de sala, conflitos de laboratório, aulas fora dos horários permitidos, indisponibilidades docentes, agrupamentos ímpares, quatro aulas consecutivas sem autorização e cargas semanais incompletas. Dessa forma, o sistema não apenas gera uma grade, mas também oferece evidências operacionais para decidir se ela pode ser publicada.

4.5 Operação da secretaria

A área da secretaria consolidou funcionalidades que antes dependeriam da interface administrativa ou de ajustes manuais dispersos. O painel operacional resume semestre, curso, último rascunho, validação, vínculos ativos, horários liberados, aulas no rascunho, aulas publicadas e rastreabilidade da última publicação. O usuário autorizado também pode tornar um semestre o atual da aplicação, o que define qual grade aparece por padrão na consulta pública.

As telas de vínculos professor-disciplina, horários permitidos e indisponibilidade docente formam a entrada operacional do gerador. Após a geração, a prévia do rascunho permite revisar visualmente a grade semanal e mover aulas por arrastar e soltar para horários liberados. Depois da publicação, a secretaria pode editar a grade oficial sem voltar ao rascunho, mantendo a mesma lógica de matriz semanal e respeitando os horários permitidos.

Esse conjunto de telas muda o papel do GHE: o sistema deixa de ser apenas uma consulta pública e passa a ser uma ferramenta de apoio à coordenação do curso. A secretaria consegue preparar dados, gerar propostas, validar restrições, publicar a grade e corrigir horários publicados dentro de um fluxo único.

4.6 Validação técnica

Os resultados também foram validados do ponto de vista técnico. A base do projeto permanece consistente, as rotas principais estão operacionais e a suíte automatizada válida

comportamentos centrais de modelos de dados, visões, coleta automatizada, importação, indisponibilidade, geração, validação, publicação e edição de horários. A Tabela 4.3 apresenta o resumo dessas verificações.

Verificação	Previsto	Realizado	Situação
<code>python manage.py check</code>	1 execução sem erro	1 execução sem erro	1/1 concluída sem problemas identificados
Testes focados de <i>scheduling</i> e secretaria	65 testes	65 testes	65/65 executados com sucesso no ambiente local
Consulta pública	Rotas principais operacionais	Rotas operacionais	Horários, matriz curricular e páginas dedicadas funcionando
Indisponibilidade docente	Coleta e edição persistentes	Coleta e edição persistentes	Acesso individual e edição pela secretaria validados
Geração automática	Rascunho gerado	Rascunho gerado	Vínculos, horários permitidos e indisponibilidades utilizados
Validação e publicação	Publicar apenas rascunho validado	Rascunho validado e publicado	Grade pública substituída após validação
Edição de grade publicada	Ajustes com controle de conflitos	Ajustes com controle de conflitos	Alteração pela secretaria validada

Tabela 4.3: *Resumo das principais verificações técnicas realizadas no GHE.*

Em termos práticos, a validação mostra que o GHE já atingiu um ponto em que sua arquitetura, seu modelo de dados, seus fluxos públicos e seus fluxos restritos podem ser analisados como resultados concretos de programa, e não apenas como proposta conceitual.

4.7 Limitações atuais

Apesar dos avanços obtidos, o projeto ainda apresenta limitações compatíveis com a etapa atual:

- Os módulos de exportação em PDF e planilha ainda permanecem como estrutura inicial;
- A funcionalidade de boletim de notificações por período ainda não foi concluída;
- A autenticação plena de professores ainda não foi desenvolvida, sendo utilizado nesta fase o acesso por identificador individual;
- Parte das inconsistências da coleta automatizada ainda exige revisão manual, especialmente em casos ambíguos do portal externo;
- O algoritmo implementado é determinístico e pode evoluir para estratégias de otimização mais sofisticadas, como heurísticas de melhoria ou algoritmo genético.

Essas limitações não anulam os resultados alcançados. Pelo contrário, delimitam com clareza o ponto em que o projeto se encontra: a base, a consulta pública, a coleta de indisponibilidade e a geração automática já funcionam; os trabalhos futuros concentram-se em exportações, comunicação, integração institucional e aprimoramento da otimização.

Capítulo 5

Considerações Finais e Sugestões para Trabalhos Futuros

Este trabalho apresentou o desenvolvimento do Gerenciador de Horários Escolares (GHE) como um sistema *web* aplicado ao contexto do IFPB, Campus Campina Grande. O objetivo central foi construir e validar uma solução capaz de organizar a consulta pública de horários, coletar indisponibilidade docente e gerar automaticamente grades acadêmicas a partir de restrições operacionais.

Os resultados obtidos mostram que a arquitetura adotada foi capaz de atender ao recorte proposto. O projeto dispõe de uma base institucional e acadêmica consistente, com suporte a múltiplas instituições e múltiplos campi, importação da grade real do campus, navegação pública por diferentes recortes, matriz curricular por curso e período, coleta restrita de indisponibilidade docente, área operacional da secretaria, geração de rascunhos, validação de conflitos, publicação da grade e edição posterior dos horários publicados.

Do ponto de vista metodológico, o trabalho evidencia a importância de tratar sistemas de horários não apenas como problemas de otimização, mas como ecossistemas de programa e operação que dependem de modelagem de dados, integridade relacional, interfaces adequadas, mecanismos de alimentação contínua da base e ferramentas de operação para a secretaria. Sem essa infraestrutura, a etapa de geração automática tende a ficar desconectada da realidade institucional.

Assim, conclui-se que o GHE alcançou um estágio funcional relevante. O sistema não se limita à consulta de horários: ele integra coleta de dados, estruturação acadêmica, restrições docentes, geração automática, validação e publicação controlada. Essa base reduz riscos para trabalhos futuros e fornece uma fundação concreta para evoluções de otimização, exportação, comunicação e integração institucional.

5.1 Sugestões para Trabalhos Futuros

Como continuidade natural deste trabalho, destacam-se as seguintes frentes:

- Aprimorar o algoritmo de geração com heurísticas de melhoria e critérios de otimização mais sofisticados;
- Evoluir os critérios de qualidade para incluir preferências docentes, melhor distribuição de aulas e aproveitamento de espaços;
- Concluir os módulos de exportação em PDF e planilha;
- Concluir o módulo de boletim de notificações por período, permitindo avisos sobre alterações na grade;
- Ampliar a coleta docente com recursos adicionais de autenticação e acompanhamento das submissões;
- Ampliar os relatórios de validação para apoiar auditoria e comparação entre diferentes rascunhos;
- Aprofundar o tratamento de grupos e turmas, especialmente em cenários de disciplinas práticas, subdivisão de aulas e uso simultâneo de recursos diferentes;
- Preparar integrações futuras com o SUAP e outros sistemas institucionais para sincronização de dados acadêmicos.

Referências Bibliográficas

[Ceschia, Gaspero e Schaerf 2022] CESCHIA, S.; GASPERO, L. D.; SCHAERF, A. Educational timetabling: Problems, benchmarks, and state-of-the-art results. *arXiv preprint arXiv:2201.07525*, 2022. Disponível em: <<https://arxiv.org/abs/2201.07525>>. Acesso em: 17 abr. 2026. 1, 2, 6

[django-environ contributors 2025] django-environ contributors. *django-environ documentation*. 2025. Disponível em: <<https://django-environ.readthedocs.io/en/latest/>>. Acesso em: 17 abr. 2026. 9

[Django Software Foundation 2025] Django Software Foundation. *Django at a glance*. 2025. Disponível em: <<https://docs.djangoproject.com/en/5.2/intro/overview/>>. Acesso em: 17 abr. 2026. 1, 4, 7

[Django Software Foundation 2025] Django Software Foundation. *Templates*. 2025. Disponível em: <<https://docs.djangoproject.com/en/5.2/topics/templates/>>. Acesso em: 17 abr. 2026. 1, 8

[Django Software Foundation 2025] Django Software Foundation. *Writing and running tests*. 2025. Disponível em: <<https://docs.djangoproject.com/en/5.2/topics/testing/overview/>>. Acesso em: 17 abr. 2026. 7

[Django Software Foundation 2025] Django Software Foundation. *Writing your first Django app, part 2*. 2025. Disponível em: <<https://docs.djangoproject.com/en/5.2/intro/tutorial02/>>. Acesso em: 17 abr. 2026. 7

[Django Software Foundation 2026] Django Software Foundation. *Security in Django*. 2026. Disponível em: <<https://docs.djangoproject.com/en/5.2/topics/security/>>. Acesso em: 14 jun. 2026. 9, 18

[Gaspero, McCollum e Schaerf 2007] GASPERO, L. D.; MCCOLLUM, B.; SCHAERF, A. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3). In: *Proceedings of the 1st International Workshop on a Scheduling Competition (SSC 2007)*. [s.n.], 2007. Disponível em: <<https://www.unitime.org/itc2007/>>. Acesso em: 17 abr. 2026. 6

[Instituto Federal da Paraíba 2026] Instituto Federal da Paraíba. *Horário Acadêmico - Campus Campina Grande*. 2026. Disponível em: <<https://horarios.ifpb.edu.br/campina/home.php>>. Acesso em: 17 abr. 2026. 1, 7

[Müller, Rudová e Müllerová 2025] MÜLLER, T.; RUDOVÁ, H.; MÜLLEROVÁ, Z. Real-world university course timetabling at the international timetabling competition 2019. *Journal of Scheduling*, v. 28, p. 247–267, 2025. Disponível em: <<https://doi.org/10.1007/s10951-023-00801-w>>. Acesso em: 17 abr. 2026. 1, 6

[OWASP Foundation 2025] OWASP Foundation. *OWASP Top 10:2025*. 2025. Disponível em: <<https://owasp.org/Top10/2025/>>. Acesso em: 14 jun. 2026. 9, 18

[Psycopg Team 2025] Psycopg Team. *Installation - Psycopg 3 documentation*. 2025. Disponível em: <<https://www.psycopg.org/psycopg3/docs/basic/install.html>>. Acesso em: 17 abr. 2026. 8

[Psycopg Team 2025] Psycopg Team. *Psycopg features*. 2025. Disponível em: <<https://www.psycopg.org/features/>>. Acesso em: 17 abr. 2026. 8

[Requests contributors 2025] Requests contributors. *Requests: HTTP for Humans*. 2025. Disponível em: <<https://requests.readthedocs.io/en/stable/>>. Acesso em: 17 abr. 2026. 4, 9, 18

[Richardson 2025] RICHARDSON, L. *Beautiful Soup Documentation*. 2025. Disponível em: <<https://www.crummy.com/software/BeautifulSoup/bs4/doc/index>>. Acesso em: 17 abr. 2026. 4, 9, 18

[Supabase 2025] Supabase. *Connecting with PSQL*. 2025. Disponível em: <<https://supabase.com/docs/guides/database/psql>>. Acesso em: 17 abr. 2026. 8

[Supabase 2025] Supabase. *Database*. 2025. Disponível em: <<https://supabase.com/docs/guides/database/overview>>. Acesso em: 17 abr. 2026. 4, 8

[The PostgreSQL Global Development Group 2025] The PostgreSQL Global Development Group. *PostgreSQL 18 Documentation: Constraints*. 2025. Disponível em: <<https://www.postgresql.org/docs/current/ddl-constraints.html>>. Acesso em: 17 abr. 2026. 1, 4, 8

[Vrielink *et al.* 2019] VRIELINK, R. A. O. *et al.* Practices in timetabling in higher education institutions: a systematic review. *Annals of Operations Research*, v. 275, n. 1, p. 145–160, 2019. Disponível em: <<https://doi.org/10.1007/s10479-017-2688-8>>. Acesso em: 17 abr. 2026. 2, 7

Apêndice A

Estrutura resumida do projeto GHE

Este apêndice apresenta uma visão resumida da organização atual do projeto, destacando seus módulos principais, rotas implementadas, comandos relevantes, casos de uso e recortes do modelo relacional usado na geração de horários.

A.1 Aplicações principais

- *institutions*: define instituição e campus;
- *academic*: reúne modalidades, cursos, períodos, semestres letivos, professores, disciplinas, salas e laboratórios;
- *scheduling*: concentra horários publicados, vínculos professor-disciplina, horários permitidos, execuções de geração, rascunhos e serviços de validação/publicação;
- *availability*: implementa campanhas, acesso individual e registro de indisponibilidade docente;
- *public*: centraliza páginas públicas, filtros, coleta automatizada, importação por etapas e prévia de rascunhos;
- *secretariat*: reúne as telas restritas de operação da secretaria;
- *exports*: mantém a estrutura inicial para exportação;
- *notifications*: mantém a estrutura inicial para boletim de notificações por período.

A.2 Rotas públicas e restritas implementadas

- `/`: página inicial do sistema;
- `/horarios/`: central de navegação de horários;
- `/horarios/cursos/<id>/`: página dedicada por curso;
- `/horarios/professores/<id>/`: página dedicada por professor;
- `/horarios/salas/<id>/`: página dedicada por sala;
- `/horarios/laboratorios/<id>/`: página dedicada por laboratório;

- `/matriz-curricular/`: visualização da matriz curricular;
- `/indisponibilidade/professores/<token>/`: formulário restrito de indisponibilidade docente por identificador individual;
- `/secretaria/`: painel operacional da secretaria;
- `/secretaria/vinculos/`: gestão de vínculos professor-disciplina;
- `/secretaria/indisponibilidades/`: revisão e edição de indisponibilidades;
- `/secretaria/horarios-permitidos/`: configuração de horários liberados para geração;
- `/secretaria/gerar/`: ação de geração de rascunho;
- `/secretaria/publicar/`: ação de publicação do rascunho validado;
- `/secretaria/horarios-publicados/`: edição da grade publicada;
- `/horarios/rascunhos/<id>/`: prévia visual do rascunho gerado.

A.3 Recorte do modelo relacional

A Tabela A.1 resume as entidades centrais usadas no domínio do GHE e o papel de cada uma na consulta pública, na coleta de indisponibilidade e na geração de horários.

No fluxo de geração, *TeachingAssignment* fornece a carga a ser alocada, *CourseAllowedSlot* limita os horários possíveis e *ProfessorUnavailabilitySlot* bloqueia faixas informadas pelos docentes. A execução é registrada em *ScheduleGenerationRun*; as aulas propostas ficam em *ScheduleDraftEntry*; após validação e publicação, os registros são gravados em *Schedule*, que alimenta a consulta pública.

A.4 Relações centrais do modelo

A Tabela A.2 complementa o recorte anterior com as principais relações usadas para manter coerência entre os dados acadêmicos, os recursos físicos, a coleta de indisponibilidade e a geração.

A.5 Casos de uso por ator

A Tabela A.3 resume os principais casos de uso do sistema por ator, complementando os diagramas apresentados no Capítulo 3.

A.6 Comandos relevantes

- `python manage.py scrape_ifpb_campina`: coleta a grade pública do portal externo;
- `python manage.py import_ifpb_campina_csv`: importa a base acadêmica e os horários para o banco;

Entidade	Papel no domínio
<i>Institution</i>	Representa a instituição mantenedora.
<i>Campus</i>	Define a unidade física e vincula cursos, professores, salas, laboratórios e semestres.
<i>Course</i>	Representa o curso e seu turno, sempre associado a um campus e modalidade.
<i>AcademicTerm</i>	Representa o semestre letivo do campus e indica qual semestre é atual na consulta pública.
<i>AcademicPeriod</i>	Representa o período do curso.
<i>Discipline</i>	Representa a disciplina vinculada a um curso.
<i>CoursePeriodDiscipline</i>	Materializa a matriz curricular ao relacionar curso, período e disciplina.
<i>Professor</i>	Representa o docente do campus.
<i>Room e Laboratory</i>	Representam recursos físicos usados nas aulas.
<i>Schedule</i>	Armazena a grade publicada consultada pelo público.
<i>TeachingAssignment</i>	Define a demanda de geração: semestre, curso, período, disciplina, professor, turma e carga semanal.
<i>CourseAllowedSlot</i>	Define quais faixas de horário o gerador pode utilizar para um curso e semestre.
<i>AvailabilityCampaign</i>	Agrupa a coleta de indisponibilidade de um campus e período letivo.
<i>ProfessorAvailability</i>	Representa a submissão individual do professor em uma campanha.
<i>ProfessorUnavailabilitySlot</i>	Registra os horários em que o professor não pode ministrar aula.
<i>ScheduleGenerationRun</i>	Registra uma execução de geração, sua situação operacional, resumo e datas.
<i>ScheduleDraftEntry</i>	Armazena os itens do rascunho antes da publicação.

Tabela A.1: Recorte textual do modelo relacional do GHE.

Origem	Destino	Finalidade
<i>Institution</i>	<i>Campus</i>	Separar unidades institucionais e permitir expansão multicampi.
<i>Campus</i>	<i>Course, Professor, Room, Laboratory</i>	Impedir mistura de cursos, docentes e espaços de campi diferentes.
<i>Course</i>	<i>AcademicPeriod</i>	Organizar a grade e a matriz curricular por período do curso.
<i>CoursePeriodDiscipline</i>	<i>Course, AcademicPeriod, Discipline</i>	Materializar a matriz curricular e evidenciar disciplinas vinculadas a cada período.
<i>TeachingAssignment</i>	<i>Professor, Discipline, AcademicTerm</i>	Definir a demanda que o gerador deve alocar em determinado semestre.
<i>CourseAllowedSlot</i>	<i>Course, AcademicTerm</i>	Restringir os horários que podem ser usados na geração.
<i>ProfessorAvailability</i>	<i>AvailabilityCampaign, Professor</i>	Representar a submissão individual de um professor em uma campanha.
<i>ScheduleDraftEntry</i>	<i>ScheduleGenerationRun</i>	Manter o rascunho isolado antes da publicação oficial.
<i>Schedule</i>	<i>Course, Professor, Room, Laboratory</i>	Alimentar a consulta pública e preservar a ocupação de recursos.

Tabela A.2: Relações centrais do modelo relacional do GHE.

Ator	Caso de uso	Resultado esperado
Comunidade acadêmica	Consultar horários por curso, professor, sala ou laboratório	Visualizar a grade pública do semestre atual por diferentes recortes.
Comunidade acadêmica	Consultar matriz curricular	Identificar disciplinas por curso e período.
Professor	Informar indisponibilidade	Registrar faixas semanais em que não pode ministrar aula.
Secretaria	Importar dados do portal público	Criar base relacional inicial para consulta e geração.
Secretaria	Configurar vínculos e horários permitidos	Preparar a entrada usada pelo gerador.
Secretaria	Gerar e revisar rascunho	Obter uma proposta de grade antes de alterar a consulta pública.
Secretaria	Validar e publicar grade	Substituir a grade pública apenas quando o rascunho não apresenta pendências impeditivas.
Secretaria	Editar grade publicada	Corrigir horários publicados dentro da área operacional.

Tabela A.3: *Resumo dos casos de uso por ator no GHE.*

- `python manage.py check`: executa checagens da aplicação Django;
- `python manage.py test`: executa a suíte automatizada do projeto;
- `python manage.py runserver`: inicia o servidor local.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande - Código INEP: 25137409
	R. Tranquílino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Ostensivo (Público)

TCC

Assunto:	TCC
Assinado por:	Mateus Sebastian
Tipo do Documento:	Dissertação
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Mateus Sebastian de Albuquerque Rocha, DISCENTE (202311210012) DE TECNOLOGIA EM TELEMÁTICA - CAMPINA GRANDE, em 02/07/2026 08:10:25.

Este documento foi armazenado no SUAP em 02/07/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1901248
Código de Autenticação: 6d2823dc05

