

Geração de Horários e Integração Entre Sistemas de Agendamento do IFSC Lages

João V. Arbegaus¹, Lucas dos S. Borges¹, Vilson Heck Jr¹

¹Instituto Federal de Santa Catarina (IFSC) - Câmpus Lages
Curso Superior de Ciência da Computação
R. Heitor Villa Lobos, 225 - 88.506-400 - Lages - SC – Brasil

joao.a2004@aluno.ifsc.edu.br, lucas.sb171@aluno.ifsc.edu.br,
vilson.junior@ifsc.edu.br

Abstract. *This article presents an integrated solution to automate the generation of academic timetables and room scheduling at IFSC Câmpus Lages. The system uses a genetic algorithm to produce timetable combinations that optimize conflicts such as teacher availability. The solution was connected to the institution's internal course-registration system through APIs and integrated directly with MRBS, enabling the automatic recording of room reservations. Web interfaces were also developed for administrative editing and public visualization of schedules. The system's integrated functionalities, by automating the generation and registration of schedules, reduce rework and replace manual processes, promoting a more systematized organization.*

Resumo. *Este artigo apresenta uma solução integrada para automatizar a geração de horários acadêmicos e o agendamento de salas no IFSC Câmpus Lages. O sistema utiliza um algoritmo genético para produzir combinações de horários que otimizam conflitos como a disponibilidade de docentes. A solução foi conectada ao sistema interno de cadastro de disciplinas por meio de APIs e integrada diretamente ao MRBS, permitindo o registro automático das reservas. Também foram desenvolvidas interfaces Web para edição administrativa e visualização pública dos horários. As funcionalidades integradas do sistema, ao automatizar a geração e o registro dos agendamentos, reduzem o retrabalho e substituem processos manuais, promovendo uma organização mais sistemática.*

1. Introdução

A elaboração de cronogramas para aulas e turmas, conhecida como *timetabling* ou *School Timetabling Problem* (STP) na literatura (Schaerf, 1999), constitui um desafio recorrente e complexo em instituições de ensino. Esse processo envolve a organização de disciplinas e aulas, considerando variáveis como a disponibilidade de professores, a carga horária das matérias, as necessidades específicas das turmas e a disponibilidade de espaços físicos, como salas de aula e laboratórios. A complexidade surge da quantidade de regras a serem seguidas, como evitar intervalos indesejados para docentes e estudantes, conciliar horários variados de professores e respeitar as limitações de oferta de disciplinas, o que torna o *timetabling* suscetível a erros e exige um significativo esforço operacional.

No Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC), Câmpus Lages, esse desafio é agravado pelo uso de múltiplos sistemas não integrados. O fluxo atual inicia-se com um sistema acadêmico interno, utilizado para registrar dados de docentes, turmas, componentes curriculares e espaços físicos, gerando um arquivo que é exportado para o software aSc TimeTables. Esse sistema terceirizado, cujo uso implica custos anuais, realiza o *timetabling*, mas exige intervenções manuais para resolver conflitos, como a impossibilidade de restringir automaticamente a alocação de um professor em três turnos no mesmo dia. Posteriormente, os horários gerados são transferidos para o Meeting Room Booking System (MRBS), que gerencia a ocupação de salas por meio da importação de arquivos ou cadastro manual. Essa fragmentação aumenta a carga de trabalho, os custos operacionais e as inconsistências, evidenciando a necessidade de uma solução mais eficiente.

Pesquisas no campo da integração de sistemas destacam que a interoperabilidade e a automação são fundamentais para incrementar a eficiência, reduzir erros e aliviar a sobrecarga operacional (Davenport, 1998), (Luftman, 2004). Nesse contexto, este trabalho propõe uma solução integrada para o problema de *timetabling* no IFSC Câmpus Lages. O objetivo é implementar um algoritmo baseado em inteligência artificial, com foco em algoritmos genéticos, para solucionar o problema de *timetabling* e integrá-lo ao sistema acadêmico interno do IFSC Lages, visando melhorar a usabilidade, reduzir a complexidade da comunicação entre as plataformas e oferecer maior flexibilidade na adaptação às restrições institucionais. Com isso, busca-se minimizar erros, otimizar recursos e aprimorar a gestão acadêmica, contribuindo para a eficiência operacional da instituição. Para alcançar esse propósito, foram definidos os seguintes objetivos específicos:

- Apresentar os sistemas utilizados no processo de geração de horários do IFSC Câmpus Lages, identificando suas funcionalidades;
- Aplicar técnicas de algoritmos genéticos para gerar automaticamente horários acadêmicos de acordo com as restrições institucionais;
- Implementar uma solução integrada que conecte o algoritmo genético, as interfaces de edição e visualização de horários e o sistema MRBS, permitindo a comunicação automatizada para geração de horários e reserva de salas.

Este trabalho foi dividido em quatro etapas que visam atingir os objetivos propostos. Na primeira etapa, será realizado um levantamento e análise dos sistemas atualmente utilizados no IFSC Câmpus Lages para o processo de geração de horários acadêmicos, com o intuito de compreender suas funcionalidades, limitações e formas de comunicação.

A segunda etapa envolve o desenvolvimento e a implementação do algoritmo genético que será responsável pela geração automática dos horários dos cursos do IFSC visando sua aplicabilidade no contexto institucional.

A terceira etapa abrangerá o desenvolvimento e a implementação das interfaces *Web* de visualização dos horários, destinadas tanto a coordenadores de disciplinas quanto a usuários comuns do sistema.

A quarta etapa será composta pela implementação da integração entre os sistemas desenvolvidos e os existentes.

Esta pesquisa se classifica, quanto à sua natureza, como aplicada; quanto à abordagem, como qualitativa; e quanto aos objetivos, como descritiva. No que se refere aos

procedimentos metodológicos adotados, trata-se de uma pesquisa bibliográfica.

Este artigo está dividido da seguinte forma: a seção 1 apresenta a introdução, abordando o problema e os objetivos do trabalho; a seção 2 apresenta os sistemas existentes em uso no IFSC Lages em 2025; a seção 3 compara trabalhos relacionados para a geração de horários utilizando algoritmos genéticos; a seção 4 mostra as etapas de desenvolvimento do trabalho, com foco na modelagem e implementação; a seção 5 expõe os resultados do novo sistema integrado; e, por fim, a seção 6 apresenta a conclusão e sugestões para trabalhos futuros.

2. Análise dos Sistemas Existentes no IFSC Lages

Esta seção descreve os sistemas utilizados pelo IFSC Câmpus Lages no processo de *timetabling*, apresentando telas das aplicações envolvidas e destacando o sistema interno desenvolvido pela própria instituição. Esse sistema é responsável pelo cadastro e gerenciamento de dados relacionados a professores, disciplinas, turmas e espaços físicos, funcionando como ponto de partida para todo o processo de geração e organização dos horários acadêmicos.

2.1. Panorama Geral do Processo de Alocação de Horários e Salas

O sistema interno foi projetado para unificar e estruturar todas as informações necessárias para a geração de horários acadêmicos. Isso possibilita o cadastro detalhado de professores (incluindo seus períodos de disponibilidade semanal) e cursos, os quais associam disciplinas (com a carga horária específica), salas de aula, professores e observações pertinentes. O sistema unifica essas informações e gera um arquivo XML padronizado, que é utilizado pelo aSc TimeTables para realizar o processo de *timetabling*.

O software, desenvolvido em PHP sem a utilização de um *framework* específico, possui banco de dados MySQL e interface *Web*, assim, assegurando a compatibilidade com a infraestrutura institucional. Servidores, como a assessoria do Departamento de Ensino, Pesquisa e Extensão (DEPE), são responsáveis por revisar e ajustar os cadastros antes da exportação e importação de dados.

Em relação à inserção de turmas e disciplinas, os coordenadores de curso são os principais responsáveis por essa função a cada semestre, bem como associar as respectivas disciplinas, cargas horárias e professores. Além disso, também registram informações sobre restrições de horários e preferências de alocação, que são fundamentais para a geração automatizada dos horários.

Após a consolidação dos dados no sistema interno, essas informações são exportadas para o aSc TimeTables, um software especializado na geração automática de horários. Essa ferramenta utiliza algoritmos de otimização para montar horários viáveis, respeitando algumas restrições impostas. O aSc permite ajustes manuais e a exportação dos resultados para o sistema de alocação de salas.

O último passo envolve o sistema de agendamento de salas, o MRBS, uma aplicação *Web* utilizada para reservar espaços físicos, como salas de aula, laboratórios e auditórios. Com base nos horários gerados pelo aSc TimeTables, os responsáveis realizam o agendamento manual das salas no MRBS, visto que, em 2025, não foi possível o cadastro automatizado de salas com base nos arquivos do software gerador de horários.

Embora os três sistemas cumpram funções distintas e complementares, a integração entre eles é feita de forma semiautomática ou manual, com necessidade de exportação, importação manual de arquivos intermediários e cadastros manuais de salas. Isso torna o processo suscetível a erros, retrabalho e consumo de tempo.

2.1.1. Fluxo de Atividades

Com o objetivo de facilitar a visualização das ações realizadas, a Figura 1 em formato de diagrama de atividades mostra os processos que constituem a geração de horários no IFSC Lages.

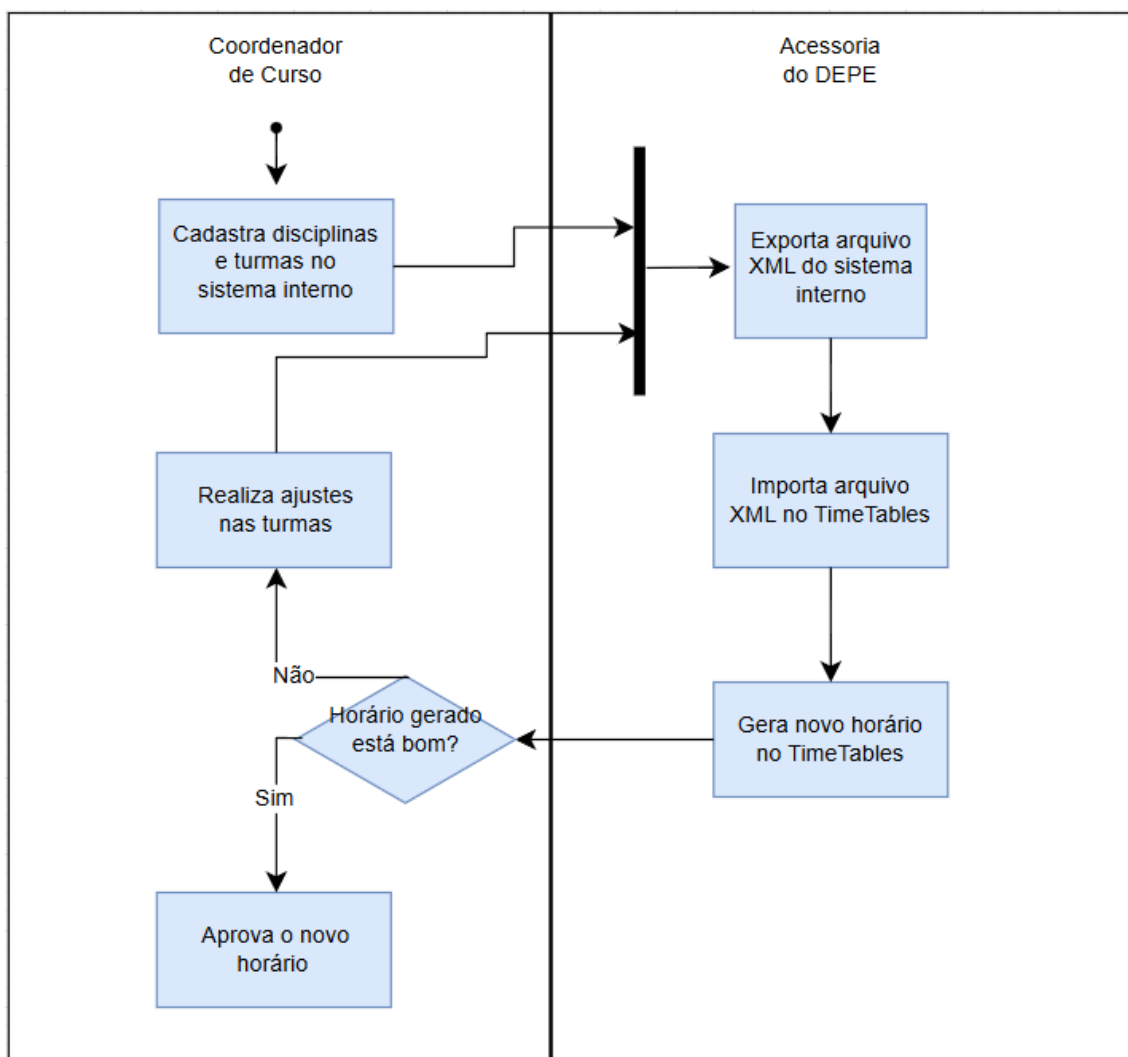


Figura 1. Fluxo de processos na geração de horários.

2.2. Interfaces

Esta subseção tem como objetivo detalhar as principais telas utilizadas pelos servidores e coordenadores de curso do IFSC nos processos de cadastro de turmas, geração de horários e alocação de salas.

2.2.1. Sistema Interno do IFSC Lages

A Figura 2 representa a tela de inclusão de docente, onde o usuário (coordenador de curso) cadastra um novo professor, informando nome, área de atuação e justificativa de eventual restrição.

Incluir novo docente:

Nome:

Área:


Justificativa: 

Figura 2. Tela de inserção de uma docente.

Além do cadastro, o usuário pode visualizar os cursos e disciplinas aos quais o docente está vinculado, como visto na Figura 3, consolidando a carga horária total e ambientes de aula. Abaixo, uma matriz de disponibilidade semanal (manhã, tarde e noite) mostra, com “V” ou “X”, os turnos em que o professor está disponível. Essa interface facilita ajustes finos nas restrições de agenda antes da exportação dos dados.

Fulano

Curso	Fase	Sigla	Disciplina	CH semanal	Distribuição semanal	Ambiente	Observações
SUP. Engenharia Mecânica	3	PGR	Programação	4	1 Quadrupla(s)	Lab. Inf. 112	se tiver lab inf maior, melhor
Tec Informática para Internet	1	BDD	Banco de Dados	4	1 Quadrupla(s)	Lab. Inf. 116	
Tec Informática para Internet	1	IPI	Introdução à Programação	6	1 Quadrupla(s)	Lab. Inf. 116	
Total:				14			

Disponibilidade:

#	Manhã				Tarde				Noite			
	M1	M2	M3	M4	T1	T2	T3	T4	N1	N2	N3	N4
Seg	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ter	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Qua	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Qui	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sex	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sab	x	x	x	x	x	x	x	x	x	x	x	x



Figura 3. Tela de visualização dos dados do docente.

A tela representada na Figura 4 exibe as disciplinas de uma turma específica (com indicativo de fase, sigla e carga horária), permitindo ao usuário selecionar o docente responsável, definir a distribuição semanal (simples, dupla, tripla), escolher o ambiente (sala ou laboratório) e inserir observações. Botões de “Customizar Período” e “Alterar Dados” dão acesso a parâmetros adicionais, enquanto o ícone de lixeira remove uma disciplina da oferta. Essa interface unifica o gerenciamento de grande parte dos aspectos necessários para o agendamento acadêmico.

Curso selecionado: Tec Informática para Internet
 Coordenador: Sem coordenador - Área: Informática - Período: Tarde

ALTERAR DADOS CUSTOMIZAR PERÍODO REMOVER CURSO

Fase	Sigla	Disciplina	Docente	CH Semestre	Distribuição semanal	Ambiente/Capacidade	Turma	Observações	Ofertar Disciplina
1	BDD	Banco de Dados	Milena Cristina Fr	80 CH	1 Quadrupla(s)	Lab. Inf. 116 (36)	Turma Completa	Observações	✓
1	CEE	Comunicação e Exg	Paula Clarice S. C	40 CH	1 Dupla(s)	Lab. Inf. 116 (36)	Turma Completa	Observações	✓
1	IPI	Introdução à Progr	João Augusto da	120 CH	1 Quadrupla(s)	Lab. Inf. 116 (36)	Turma Completa	Observações	✓
1	SEC	Sistemas e Conecti	Fernando Sagaz	80 CH	1 Quadrupla(s)	Lab. Inf. 116 (36)	Turma Completa	Observações	✓
2	DES	Desenvolvimento o	Fernando Sagaz	40 CH	1 Dupla(s)	Lab. Inf. 114 (29)	Turma Completa	Observações	✓
2	DES	Desenvolvimento o	Fernando Sagaz	80 CH	1 Quadrupla(s)	Lab. Inf. 114 (29)	Turma Completa	1 quádrupla e 1 dupla	✓
2	SEG	Segurança da Infor	Substituto Vison	40 CH	1 Dupla(s)	Lab. Inf. 114 (29)	Turma Completa	Observações	✓

Figura 4. Tela de visualização e adição dos dados do curso.

2.2.2. aSc TimeTables

O software aSc TimeTables é utilizado como parte do processo de geração de horários no IFSC Lages. Este sistema é desenvolvido e comercializado pela empresa aSc Applied Software, sediada na Eslováquia, e está disponível no site oficial: <https://www.asctimetables.com> (aSc Applied Software, 2025).

A Figura 5 apresenta a interface principal do software aSc Timetables, onde, após a exportação dos dados em formato XML pelo sistema interno, o arquivo gerado é importado no aSc, que gera uma grade horária com base nas disciplinas, professores e restrições definidas previamente.

Cada linha na grade representa uma turma (ex: CC1, CC3) e as colunas representam os dias da semana e períodos letivos. Os blocos coloridos indicam as aulas atribuídas, possibilitando a identificação das disciplinas ao longo da semana.

Apesar de sua utilização, o aSc Timetables apresenta limitações significativas que comprometem a eficiência e a flexibilidade do processo. Um dos principais problemas observados é a inexistência de uma funcionalidade nativa para limitar a quantidade de turnos que um professor pode lecionar por dia — por exemplo, restringir um docente a atuar em no máximo dois períodos (manhã e tarde) ao longo do dia.

Outro ponto crítico é a integração semiautomática com os demais sistemas. Tanto a importação dos dados via XML quanto a exportação dos horários consolidados depen-

dem de intervenção manual, o que aumenta o risco de inconsistências e erros.

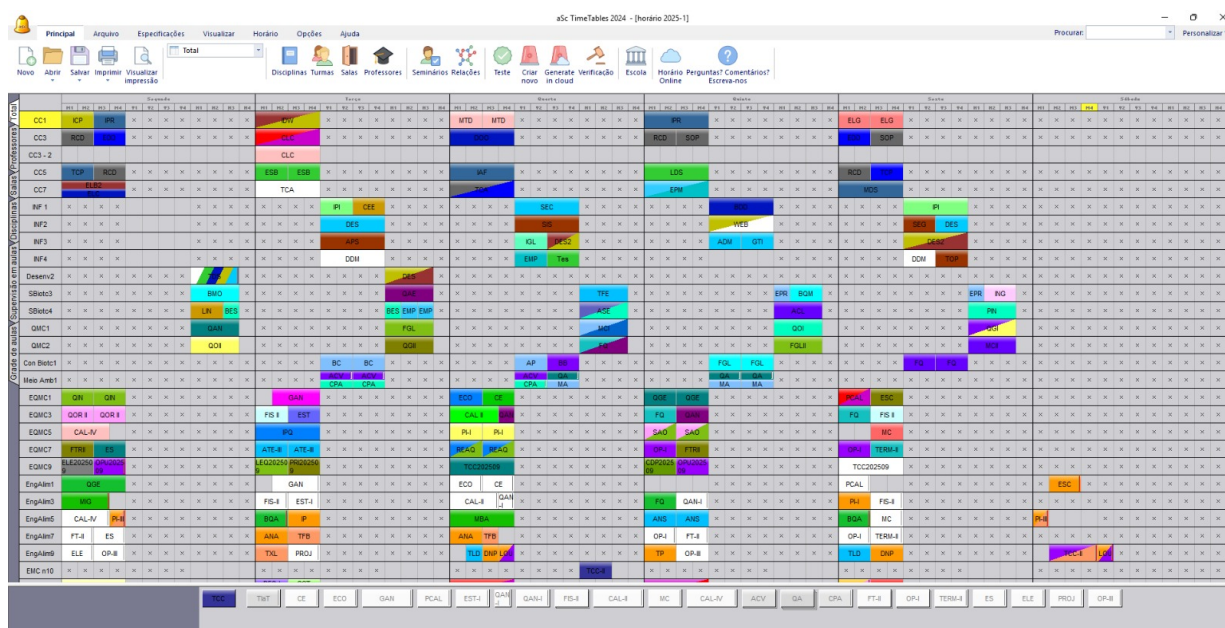


Figura 5. Tela de Visualização e Manipulação de Horários no ASC Timetables.

2.2.3. Sistema de Agendamento de Salas – MRBS

O MRBS é o sistema atualmente utilizado pelo IFSC Câmpus Lages para o agendamento de salas de aula, laboratórios e demais espaços físicos institucionais. Trata-se de uma aplicação *Web* de código aberto, desenvolvida e mantida pelo MRBS Team (2025), com interface simples e funcional, permitindo que servidores e docentes reservem horários específicos para diferentes ambientes, registrando informações como turma, disciplina ou atividade vinculada.

A Figura 6 ilustra a visualização diária do MRBS, exibindo a distribuição das reservas por sala ao longo do dia. Nessa interface, cada coluna representa um ambiente físico e cada bloco verde corresponde a uma reserva confirmada. A navegação pode ser alternada entre modos diário, semanal e mensal, favorecendo consultas rápidas ou planejamento de períodos maiores.

Apesar de sua praticidade para operações manuais, como reservar uma sala para uma aula isolada ou evento, o MRBS não foi projetado especificamente para atender às necessidades complexas de instituições de ensino. Sua estrutura é adequada para agendamentos pontuais, mas apresenta limitações quando aplicada à organização completa da malha de horários letivos, que exige controle simultâneo de múltiplas turmas, professores, laboratórios e restrições pedagógicas.

Uma das principais limitações é a ausência de integração nativa com sistemas capazes de gerar automaticamente horários acadêmicos. Embora o aSc Timetables ofereça ferramentas de exportação de dados, no semestre letivo de 2025 não foi possível estabelecer a comunicação entre o aSc e o MRBS devido à incompatibilidade entre os formatos de

arquivo gerados, impossibilitando o fluxo automático de dados, resultando em registros manuais no MRBS de cada reserva correspondente aos horários previamente planejados, aumentando o esforço operacional e a suscetibilidade a erros.

Em função dessas limitações, um dos objetivos deste trabalho é o avanço na automação e integração entre os sistemas, reduzindo tarefas manuais e eliminando o re-trabalho entre a geração dos horários e o agendamento dos espaços físicos.

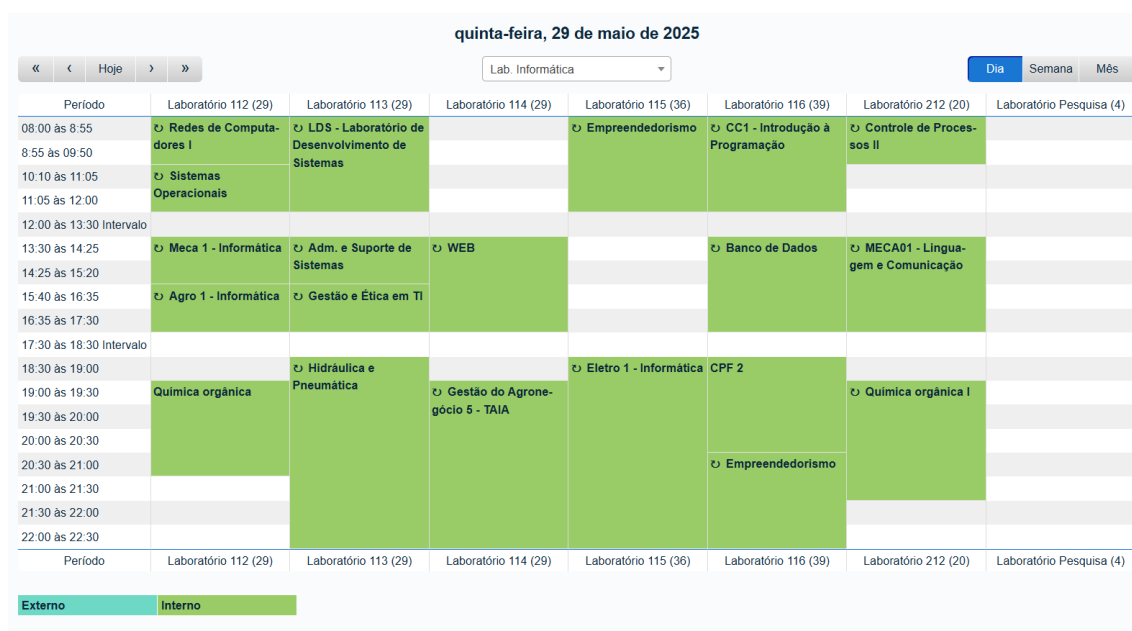


Figura 6. Interface do MRBS com visualização diária das reservas por sala.

3. Trabalhos Correlatos

Para o levantamento bibliográfico, foi realizada uma busca sistemática no *Google Acadêmico* com o objetivo de identificar estudos que aplicam Algoritmos Genéticos (AG) a problemas de restrições e ao *timetabling*. A seleção considerou artigos que exploram tanto a modelagem de *hard* e *soft constraints* quanto propostas de integração de sistemas e otimização de cronogramas acadêmicos, garantindo uma visão abrangente das abordagens, avanços e lacunas na área.

Rebonatto et al. (2025) apresentam uma abordagem baseada em Algoritmos Genéticos voltada à geração de horários acadêmicos em instituições que possuem currículos com disciplinas compartilhadas entre diferentes cursos. O trabalho destaca o desafio da simultaneidade docente e dos conflitos gerados quando múltiplas turmas dependem das mesmas disciplinas ou dividem professores em comum. A solução proposta busca minimizar tais conflitos por meio da modelagem explícita dessas interdependências no processo evolutivo, permitindo que o algoritmo considere tanto restrições rígidas, como incompatibilidades de horários, quanto restrições flexíveis, relacionadas à distribuição equilibrada das aulas.

O estudo de Xu e Petrie (2012) apresenta os *Self-Learning Genetic Algorithms* para problemas de satisfação de restrições, combinando técnicas de autoaprendizado e

pré-processamento automático. Inicialmente, o algoritmo é treinado em instâncias de pequeno porte, de modo a identificar quais métodos de filtragem de restrições são mais eficazes. Em seguida, o AG aplica esse conhecimento na configuração de *solvers* para problemas de maior escala, reduzindo significativamente o tempo de busca. Operadores especializados de cruzamento e mutação asseguram que novas soluções respeitem os limites das restrições, mantendo diversidade populacional e evitando ótimos locais. Os resultados demonstram ganhos de desempenho tanto na rapidez de convergência quanto na qualidade das soluções encontradas. Essa abordagem híbrida, ao automatizar a seleção de parâmetros de pré-processamento, simplifica a parametrização de *solvers* complexos. Sua aplicabilidade a cenários como o *timetabling* ressalta o potencial dos AGs auto-aprendizes em contextos com múltiplas variáveis e rígidas restrições.

O trabalho de Zhang et al. (2014) destaca as vantagens dos Algoritmos Genéticos em cenários com restrições rigorosas, enfatizando a exploração de uma população diversificada de soluções para ampliar o espaço de busca. Ao permitir a inclusão de soluções inviáveis durante a evolução, o método aproveita informações únicas desses indivíduos para guiar cruzamentos que possam gerar descendentes mais promissores. Esse uso estratégico de soluções inviáveis ajuda a evitar a estagnação em ótimos locais, enquanto a diversidade mantida na população previne perdas prematuras de variabilidade. Além disso, o artigo demonstra que essas abordagens elevam o desempenho geral, resultando em pesquisas mais profundas e robustas. A pesquisa valida que a combinação de diversidade, aproveitamento de soluções inviáveis e recombinação inteligente de segmentos genéticos fortalece a capacidade dos AGs de resolver problemas de otimização com múltiplas restrições de forma eficaz.

O trabalho de Sani e Yabo (2016) ilustra a eficácia dos Algoritmos Genéticos na solução de problemas de *timetabling*, destacando sua robustez ao lidar com espaços de busca altamente complexos. Ao representar horários como cadeias cromossômicas, o AG oferece flexibilidade para incorporar diferentes tipos de restrições — como conflitos de exames, limitações de sala e cargas horárias — sem necessidade de modelagem excessivamente rígida. Os operadores de cruzamento e mutação permitem explorar amplamente o espaço de soluções, enquanto mecanismos de seleção garantem a preservação de esquemas promissores. Os resultados mostram que essa abordagem reduz significativamente a ocorrência de conflitos e melhora a cobertura das janelas livres, ao mesmo tempo em que mantém um desempenho computacional aceitável. Além disso, o estudo evidencia a capacidade dos AGs de adaptar-se a variações nos requisitos institucionais, tornando-os especialmente indicados para cenários acadêmicos com demandas diversificadas e dinâmicas. Por fim, a aplicação proposta por Sani e Yabo reforça o valor dos Algoritmos Genéticos como ferramenta flexível e eficaz para o agendamento automático de cronogramas.

O estudo de Thakare et al. (2020) demonstra que os Algoritmos Genéticos podem transformar a geração de cronogramas acadêmicos ao codificar turmas, professores e salas em cromossomos e evoluir soluções que minimizam conflitos e respeitam capacidades, reduzindo drasticamente a necessidade de ajustes manuais. Comparado a métodos tradicionais, o AG mostrou ganhos substanciais em tempo de processamento e qualidade dos horários gerados, com redução expressiva de erros humanos e iterações manuais. A flexibilidade da representação cromossômica permite incorporar regras complexas de restrição

com facilidade, e o processo evolutivo ajusta progressivamente as soluções, resultando em cronogramas quase livres de falhas e cada vez mais adequados às necessidades institucionais. Esses resultados reforçam o potencial dos AGs para automatizar completamente o *timetabling* em larga escala, atendendo a rigorosas condições de restrição e melhorando a satisfação de docentes e alunos.

O trabalho de Bianchini e Silva (2020) propõe a aplicação de um Algoritmo Genético Distribuído para resolver o problema de *timetabling* no IFSC Lages, com o objetivo de gerar quadros de horários sem violar as restrições institucionais. A proposta se destaca pela arquitetura híbrida, que permite o processamento centralizado ou distribuído, dividindo os cursos em conjuntos e alocando-os a diferentes máquinas para acelerar o tempo de execução. Além disso, o sistema implementa um pós-processamento baseado em busca em profundidade limitada para resolver conflitos remanescentes. Os testes demonstraram que a solução distribuída foi significativamente mais eficiente, atingindo resultados em média 2,7 vezes mais rápidos do que o modelo centralizado. Essa abordagem valida a eficácia dos algoritmos genéticos em ambientes com múltiplas restrições e reforça sua aplicabilidade prática em instituições de ensino.

Além disso, o estudo de Albino (2022), desenvolvido na UFSC, aplica Algoritmos Genéticos ao problema de alocação de horários e professores no ensino fundamental, com ênfase na experimentação de parâmetros (taxas de cruzamento e mutação), uso de elitismo e testes de uma variante híbrida com busca local. A representação cromossômica modela a grade semanal por professor, e a função de aptidão penaliza conflitos rígidos (choques de professor/turma). Os resultados indicam que o AG encontra soluções factíveis em tempo razoável, que o elitismo melhora a convergência quando balanceado com diversidade e que melhorias na população inicial nem sempre reduzem o tempo total; o trabalho também sugere paralelização e estudos com dados reais como próximos passos.

Com base nos trabalhos analisados, observa-se que os Algoritmos Genéticos (AG) se mostram uma solução válida e eficaz para o problema de *timetabling*, especialmente em contextos acadêmicos com múltiplas variáveis e restrições rígidas. Estudos como os de Thakare et al. (2020), Sani e Yabo (2016) e Bianchini e Silva (2020) demonstram que os AGs são capazes de gerar cronogramas de qualidade, reduzindo conflitos, otimizando o tempo de processamento e se adaptando a diferentes cenários institucionais. Sua flexibilidade na representação de soluções e a capacidade de evoluir populações de horários com base em critérios definidos permitem que se alcancem resultados satisfatórios mesmo em espaços de busca altamente complexos. Assim, a escolha dos AGs como base para este trabalho justifica-se tanto pelo suporte teórico quanto pelos resultados práticos observados em implementações anteriores.

A seguir, apresenta-se um comparativo entre trabalhos relacionados que utilizam Algoritmos Genéticos na geração de cronogramas acadêmicos. O quadro destaca aspectos como o nível de aplicação, as técnicas utilizadas, o tipo de restrições tratadas, permitindo situar a proposta deste trabalho em relação às soluções já existentes.

Autores	Nível de Aplicação	Técnica Principal	Restrições Tratadas
Thakare et al. (2020)	Universitário	Algoritmo Genético	Professores, salas, conflitos de horário
Sani e Yabo (2016)	Universitário / Médio	Algoritmo Genético	Conflitos de exame e sobreposição
Zhang et al. (2014)	Genérico (restrições)	AG com soluções inviáveis	Restrições lineares e não-lineares
Bianquini & Hofman (2020)	Universitário (IFSC)	AG distribuído + heurística	professores, turmas, preferências
Albino (2022)	Ensino Fundamental	AG híbrido	Conflitos professor/turma, número e preferência de aulas
Projeto proposto (2025)	Universitário (IFSC)	Algoritmo Genético	Professores, turmas, turnos, preferências, salas

Quadro 1. Comparativo de soluções para *timetabling*

Observa-se, de modo geral, que os trabalhos analisados consolidam o uso de Algoritmos Genéticos como uma abordagem robusta para problemas de *timetabling*, destacando-se pela capacidade de explorar espaços de busca complexos e lidar com múltiplas restrições simultâneas. Apesar das diferenças de escopo e ambiente de aplicação, todos contribuem para o entendimento de estratégias que melhoram a eficiência da busca evolutiva, como a priorização de elementos mais restritivos, a adoção de operadores especializados e a combinação de heurísticas complementares.

Essas contribuições teóricas e práticas serviram como base para a concepção deste trabalho, que adota os princípios consolidados dos AGs para a geração de horários acadêmicos, mas amplia seu escopo ao propor uma integração direta com o sistema institucional do IFSC Câmpus Lages. Assim, o presente projeto não busca apenas avançar na qualidade da otimização evolutiva, mas também demonstrar como essas técnicas podem ser incorporadas a fluxos reais de gestão acadêmica, reduzindo etapas manuais e promovendo maior consistência entre cadastro e cronograma.

4. Desenvolvimento

Esta seção apresenta o desenvolvimento do sistema proposto para a geração de horários escolares utilizando um algoritmo genético, implementado com base na biblioteca *Distributed Evolutionary Algorithms in Python* (DEAP). O objetivo é detalhar a construção do algoritmo, incluindo a modelagem dos indivíduos, dos operadores genéticos utilizados e a justificativa para a escolha da biblioteca DEAP. Além disso, aborda a integração dos sistemas desenvolvidos com o conjunto de sistemas utilizados na instituição, descrevendo como os dados são processados e os resultados incorporados aos processos institucionais. A seção está organizada em quatro subseções: Algoritmo de Geração de Horários, que detalha a modelagem do algoritmo genético; Integração entre os Sistemas, que discute a interoperabilidade entre o sistema proposto e o sistema existente na instituição; Apresentação da Interface, que aborda o desenvolvimento das telas de edição e visualização de horários, enfatizando recursos de validação, detecção de conflitos e usabilidade; e Exportação de Dados para o MRBS, que descreve o processo de consolidação automática dos horários aprovados diretamente no banco de dados do sistema de reservas.

4.1. Algoritmo de Geração de Horários

Os algoritmos genéticos são técnicas de busca e otimização inspiradas nos princípios da seleção natural e da genética, propostas inicialmente por John H. Holland em seu trabalho seminal *Adaptation in Natural and Artificial Systems* (Holland, 1992). Holland define os AGs como métodos adaptativos que simulam a evolução de populações, utilizando mecanismos como reprodução, mutação e seleção para resolver problemas complexos onde soluções analíticas tradicionais podem ser inviáveis.

Uma definição mais moderna, apresentada por Eiben e Smith (2015), descreve os AGs como algoritmos evolucionários que operam sobre uma população de soluções candidatas, representadas como estruturas codificadas (cromossomos), evoluindo-as iterativamente por meio de operadores inspirados na biologia. Esses algoritmos são utilizados em otimização combinatorial, aprendizado de máquina e problemas de escalonamento, devido à sua capacidade de explorar eficientemente espaços de busca complexos e encontrar soluções de boa qualidade, mesmo em cenários com restrições rígidas.

Os componentes principais de um algoritmo genético incluem:

- **Indivíduos:** Representam soluções candidatas para o problema. No contexto do sistema proposto, cada indivíduo é uma lista de genes, onde cada gene codifica uma alocação de aula, incluindo informações como dia, período, disciplina, turma, professor, sala e duração, conforme descrito na subseção 4.1.1.
- **Função de Aptidão (*fitness*):** Avalia a qualidade de cada indivíduo com base em critérios específicos do problema. No sistema desenvolvido, a função de aptidão penaliza violações de restrições, como conflitos de professores e salas, transformando a penalidade total em um valor de aptidão por meio de uma função, como detalhado na subseção 4.1.3. Essa abordagem garante que soluções viáveis tenham maior aptidão, enquanto soluções inviáveis, ainda assim, contribuam para a busca, alinhando-se com Zhang et al. (2014).
- **População:** Conjunto de indivíduos que evolui ao longo das gerações. No sistema proposto, a população inicial é composta por indivíduos gerados semi-aleatoriamente por uma função que respeita as restrições de disponibilidade de horários das turmas e também de professores.
- **Operadores Genéticos:** Incluem seleção, cruzamento e mutação, que guiam a evolução da população.
 - **Seleção:** Escolhe indivíduos para reprodução com base em sua aptidão. O sistema utiliza a seleção por roleta, assim como o utilizado por Bianchini e Silva (2020).
 - **Cruzamento:** Combina genes de dois indivíduos para gerar descendentes. Um operador é empregado, trocando segmentos entre indivíduos, promovendo a recombinação de alocações de horários.
 - **Mutação:** Introduce variações aleatórias para manter a diversidade da população e evitar mínimos locais conforme descrito por Xu e Petrie (2012).
- **Critério de Parada:** Define quando o algoritmo deve terminar. No sistema proposto, o algoritmo executa um número predefinido de gerações ou até que a aptidão máxima atinja 1.0, indicando uma solução sem violações.
- **Elitismo:** É a técnica que garante que uma quantidade dos melhores indivíduos de uma geração seja mantida para a próxima, evitando a perda das melhores soluções e assegurando que o melhor indivíduo da nova geração seja, no mínimo, tão bom quanto o da geração anterior (Capaverde, 2025).

Esses componentes, implementados com a biblioteca DEAP, conforme descrito na subseção 4.1.7, formam a base do algoritmo genético desenvolvido que visam a geração de horários que atendem às restrições institucionais. A combinação de uma representação flexível, uma função de aptidão detalhada e operadores genéticos personalizados reflete

os princípios fundamentais dos AGs descritos por Holland (1992) e adaptados às necessidades modernas de otimização, conforme discutido por Eiben e Smith (2015).

Dessa forma, o objetivo principal do algoritmo é gerar horários para cada turma dos *cursos*¹ existentes e que não violem as seguintes restrições:

- Turmas com duas aulas no mesmo dia e período;
- Aulas alocadas fora da disponibilidade das turmas;
- Salas utilizadas simultaneamente por turmas diferentes;
- Conflito de horários entre professores;
- Indisponibilidade dos professores.
- Turnos excessivos de professores.

As restrições relacionadas a turmas com aulas no mesmo dia e aulas fora da disponibilidade da turma foram solucionadas durante a modelagem do algoritmo genético e de seus operadores, conforme será detalhado ao longo desta seção. Dessa forma, a função de aptidão não precisa verificar sua ocorrência representando um ganho de desempenho durante a execução do algoritmo.

As restrições que ainda necessitam de verificação pela função *fitness* são as de conflito de horários entre professores, que ocorre quando mais de uma aula é atribuída a um docente em um mesmo período e dia, a indisponibilidade de professores, ocorrendo quando um professor não poderia dar aula naquele período e a de turnos excessivos que acontece quando um professor leciona durante a manhã, tarde e noite em um mesmo dia. Essas restrições foram classificadas como *hard*, pois sua ocorrência é bastante indesejável. A restrição de salas sendo ocupadas simultaneamente foi classificada como *soft*, pois há casos especiais em que duas turmas têm aulas em uma mesma sala ao mesmo tempo sem ser indesejável, apesar de que a maioria dos casos não ser dessa forma.

4.1.1. Composição dos Indivíduos

No contexto do algoritmo genético desenvolvido, cada indivíduo representa uma possível solução para o problema de geração de horários, ou seja, um conjunto de alocações de aulas que buscam respeitar as restrições do problema. Um indivíduo é modelado como uma lista de genes, onde cada gene corresponde a uma instância de uma aula a ser alocada no horário. Cada gene é um dicionário que contém as seguintes informações:

- **lessonid**: Identificador único da aula, extraído dos dados enviados pelo Sistema Interno do IFSC Lages.
- **day**: Dia da semana em que a aula está alocada (e.g., "Monday", "Tuesday", ..., "Saturday"), mapeado para índices inteiros (0 a 5) para facilitar o processamento.
- **period**: Período do dia em que a aula vai começar após a alocação (valores de 0 a 11, correspondendo a 12 períodos diários).
- **subjectid**: Identificador da disciplina associada à aula.
- **classid**: Identificador da turma associada à aula.

¹Neste contexto, o termo *curso* refere-se ao conjunto de turmas de um mesmo programa de ensino (por exemplo, Ciência da Computação 1 e Ciência da Computação 3), cujas aulas devem ser organizadas em um cronograma individual mas possuem as mesmas restrições de horários por exemplo.

- **teacherid**: Lista de identificadores dos professores associados à aula, permitindo múltiplos professores, se necessário.
- **classroomid**: Identificador da sala de aula alocada.
- **duration**: Duração da aula em períodos consecutivos.

4.1.2. Inicialização dos Indivíduos

A inicialização ocorre a partir dos dados recebidos por meio de uma chamada à *API*, na qual o sistema interno do IFSC envia as informações das turmas e aulas cadastradas. O servidor que executa o algoritmo genético processa esses dados e, durante a execução, constrói o indivíduo de forma sequencial, atribuindo a cada gene (aula) um dia e um período de maneira semi-aleatória, respeitando as restrições de disponibilidade de turmas, professores e salas.

Essas restrições são representadas por matrizes binárias (0 e 1), nas quais as linhas correspondem aos dias da semana — de segunda a sexta-feira — e as colunas aos períodos (0 a 11). Os valores iguais a 1 indicam horários disponíveis, semelhante ao ilustrado na Figura 3. Essa representação cromossômica é inspirada em trabalhos como Sani e Yabo (2016) e Thakare et al. (2020), que destacam a flexibilidade de codificar horários como cadeias cromossômicas para lidar com restrições complexas, como conflitos de professores, turmas e salas.

Durante a inicialização, define-se uma ordem determinística das turmas, priorizando-se, dentro de cada uma, as aulas de maior duração. Essa estratégia tem como objetivo reduzir a ocorrência de lacunas ao final da alocação, uma vez que aulas mais longas apresentam maior dificuldade de encaixe quando restam apenas períodos fragmentados. Para cada aula, identificam-se os dias e períodos viáveis segundo a matriz de disponibilidade da turma e a duração da aula. Entre esses períodos, são mantidas apenas as opções em que a duração completa da aula não ultrapassa o limite disponível no dia, e nas quais o professor não apresenta indisponibilidade de horário. Dentre as alternativas válidas, o algoritmo escolhe aleatoriamente uma posição entre as opções válidas buscando alocar no início dos períodos daquela turma.

Caso uma aula não possa ser alocada no dia corrente sem violar a disponibilidade (por exemplo, quando a soma do período inicial com a sua duração extrapola o limite do dia para a turma), avança-se para o próximo dia disponível e repete-se o procedimento de busca. Esse processo resulta em indivíduos distintos a cada inicialização devido ao componente aleatório na seleção entre alternativas.

Embora não tenham sido realizados testes comparativos específicos entre diferentes estratégias de ordenação inicial, observou-se empiricamente que a priorização de aulas mais longas contribui para uma menor incidência de conflitos nas soluções iniciais, favorecendo a convergência do algoritmo ao longo das gerações. Testes adicionais, como a comparação com ordenações aleatórias ou inversas, em casos onde há mais aulas de curta duração, podem ser considerados em trabalhos futuros para uma avaliação mais aprofundada dessa estratégia.

A Figura 7 ilustra a estrutura genética de um indivíduo no algoritmo, em que cada gene, representado por G1, G2, G3, e assim por diante, corresponde a uma aula

dentro do cronograma, cada um com as informações apresentadas na subseção 4.1.1. Por conta de cada turma possuir diferentes durações e quantidades de dias com aulas para serem alocadas, a posição dentro da lista não corresponde exatamente a um dia ou período específico, essas informações estão contidas dentro de cada gene, mas a sequência de turmas é a mesma entre cada solução do algoritmo, pois na operação de cruzamento as trocas entre as soluções não podem gerar horários com aulas faltando ou incoerentes.

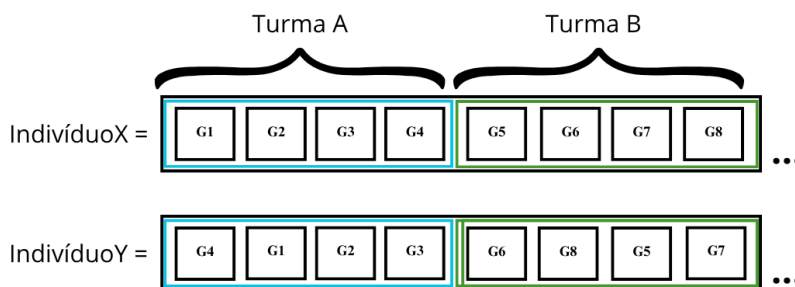


Figura 7. Representação visual do indivíduo.

4.1.3. Função de Aptidão (*Fitness*)

A função de aptidão é o componente central do algoritmo genético e segundo Liden (2008) é a maneira utilizada pelos algoritmos genéticos para determinar a qualidade de um indivíduo como solução de um determinado problema. O objetivo é maximizar o valor de aptidão, que no algoritmo atual é calculada conforme a Equação 1:

$$\text{fitness} = \frac{5,0}{5,0 + \left(\frac{\sum_i p_i}{(N_c \times 15)} \right)} \quad (1)$$

onde:

- p_i representa cada penalidade individual por violação de restrições
- $\sum_i p_i$ é o somatório total de todas as penalidades do indivíduo
- N_c é o número total de cursos no problema
- $(N_c \times 15)$ normaliza as penalidades pela dimensão do problema

Esta abordagem combina uma estrutura de referência fixa como em Santos (2016) e com ponderação flexível a exemplo de Chikwiriro et al. (2014), visando evitar quedas bruscas no *fitness*. O valor adotado de 15 representa uma estimativa controlada da ocorrência simultânea de conflitos de sala e de conflitos de professores (respectivamente penalidades de 5 e 10), que em determinados cenários são parcialmente inevitáveis.

Em contextos nos quais existem turmas com aulas concomitantes ou restrições institucionais rígidas, alguns conflitos tornam-se impossíveis de serem totalmente eliminados, mesmo em soluções próximas do ótimo. Dessa forma, o valor de 15 busca simular uma pequena quantidade tolerável dessas quebras de restrições por turma, permitindo que o algoritmo continue explorando o espaço de busca sem penalizar excessivamente soluções potencialmente viáveis.

Essa transformação faz com que indivíduos com menores penalidades tenham valores de aptidão próximos de 1, enquanto soluções com muitas violações de restrições recebem valores próximos de 0. Dessa forma a penalidade total é composta por várias penalidades específicas, associadas às seguintes restrições:

- **Conflitos de professores:** Penalidade de 10 pontos para cada período em que um professor está alocado em mais de uma aula simultaneamente no mesmo dia.
- **Conflitos de salas:** Penalidade de 5 pontos para cada período em que uma sala está ocupada por mais de uma aula.
- **Indisponibilidade de professores:** Penalidade de 15 pontos para cada período em que um professor é alocado em um horário em que está indisponível, conforme a matriz de `timeoff`.
- **Turnos excessivos de professores:** Penalidade de 25 pontos por turno adicional (manhã, tarde ou noite) além do limite de dois turnos por dia para um mesmo professor.

4.1.4. Seleção

Após avaliados, os indivíduos são submetidos a um esquema de seleção composto por duas estratégias complementares:

1. **Elitismo:** uma fração fixa dos melhores indivíduos é preservada e copiada diretamente para a próxima geração. Essa salvaguarda evita a perda de boas soluções e acelera a convergência.
2. **Roleta (*fitness-proportionate*):** os demais indivíduos são escolhidos de forma estocástica, com probabilidade proporcional à sua qualidade. Seja \mathcal{P} a população e $f(i)$ o *fitness* do indivíduo $i \in \mathcal{P}$. A probabilidade de seleção é

$$\Pr(i) = \frac{f(i)}{\sum_{j \in \mathcal{P}} f(j)}.$$

Desse modo, todos mantêm chance de reprodução, porém soluções superiores são favorecidas.

4.1.5. Cruzamento

Os indivíduos selecionados são pareados e recombinaados para produzir descendentes. Emprega-se um operador que preserva a coesão por turma, trocando blocos que possuem o tamanho de uma turma. Essa estratégia preserva subestruturas úteis (por exemplo, um arranjo de aulas já consistente dentro de uma turma), ao mesmo tempo em que possibilita a exploração de novas combinações entre turmas distintas. Essa estratégia foi adotada para que pontos de cortes no meio de uma turma não existissem, pois nesse caso se a ordem de distribuição de aulas entre os dois pais não for a mesma as aulas poderiam ser perdidas ou duplicadas dependendo de como foram formados os descendentes do cruzamento.

4.1.6. Mutação

Para manter diversidade e escapar de ótimos locais, aplica-se mutação com probabilidade controlada. O operador realiza trocas de dia/período entre genes compatíveis (mesma turma e mesma duração), alterando pontualmente o posicionamento de aulas sem violar a integridade da representação. Trabalhos como Bianquini e Silva (2020) e Vasconcelos (2021) também apresentaram os mesmos procedimentos para a operação de mutação, visto que realizando-se dessa forma, após a operação, horários com quebras de restrições relacionadas a aulas fora da disponibilidade de turmas ou turmas com duas aulas no mesmo dia e período são evitadas.

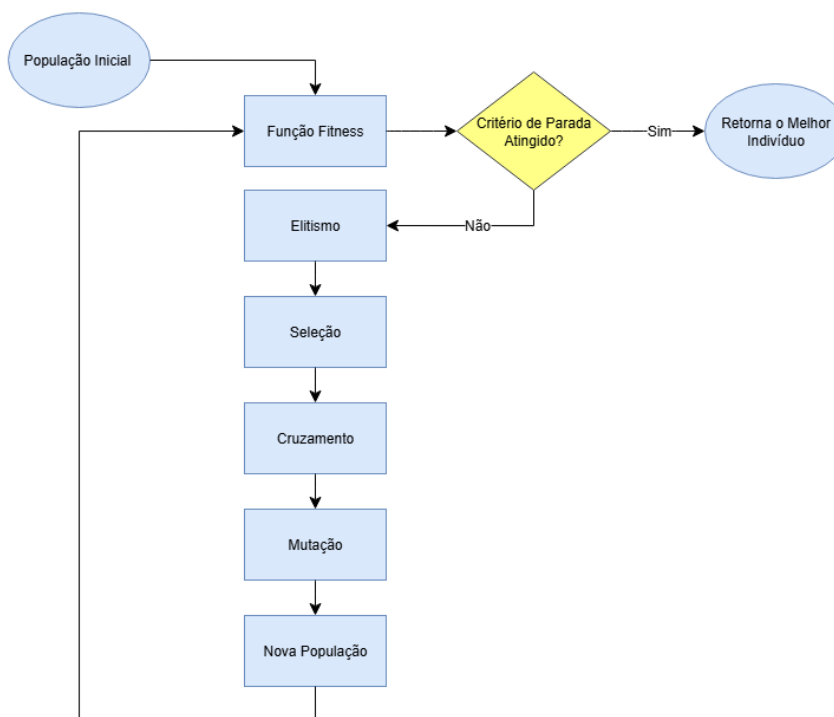


Figura 8. Diagrama de atividades do Algoritmo Genético.

Tendo em vista as partições do algoritmo já citadas, o funcionamento geral ocorre a partir da população inicial (Seção 4.1.2), então o algoritmo evolui os indivíduos por sucessivas gerações, seguindo o fluxo ilustrado na Figura 8: avaliação (função *fitness*), seleção, cruzamento, mutação e formação de nova população, até o atendimento de um critério de parada.

4.1.7. Definição da Biblioteca

Considerando os trabalhos de Xu e Petrie (2012), Zhang et al. (2014), Sani e Yabo (2016), Thakare et al. (2020), e Bianquini e Silva (2020), optou-se por utilizar a biblioteca DEAP devido à sua robustez, flexibilidade e suporte a operadores genéticos personalizados, que são essenciais para lidar com as restrições complexas do problema de *timetabling*. A escolha do DEAP foi motivada por oferecer um design de algoritmo explícito, estruturas de

dados transparentes e fácil paralelismo, permitindo que os usuários controlem todos os aspectos dos processos evolutivos, conforme descrito na sua documentação (Fortin et al., 2012). Essa flexibilidade, aliada a um conjunto substancial de exemplos, facilita a prototipagem rápida e o teste de algoritmos genéticos, o que foi crucial para a personalização do algoritmo às especificidades do problema de geração de horários do IFSC Lages. Adicionalmente, o DEAP proporciona inúmeras vantagens, incluindo métodos de seleção integrados, várias técnicas de pesquisa e uma estrutura intuitiva que permite manipular facilmente os mecanismos evolutivos, tornando-o acessível tanto para pesquisadores iniciantes quanto experientes em computação evolutiva (De Lima et al., 2022). A biblioteca é bastante referenciada em problemas de otimização como no *review* realizado por Kim e Yoo (2019) que destacou seu uso para desenvolvimento de algoritmos genéticos independentemente do nível de conhecimento do usuário, o que reforçou sua adoção ao contexto deste trabalho.

4.2. Integração entre os Sistemas

A arquitetura proposta prevê que o algoritmo genético seja incorporado como um módulo adicional ao sistema de cadastro de professores e disciplinas, consumindo os dados já cadastrados e processando-os de acordo com as restrições institucionais definidas. O objetivo é que, após a execução do algoritmo, o próprio sistema seja capaz de exibir os horários gerados em sua interface administrativa, permitindo a análise, validação e ajustes por parte dos usuários responsáveis. Dessa forma, a geração de horários deixa de depender de ferramentas externas e passa a ocorrer de forma centralizada, dentro do ambiente já familiar aos servidores técnico-administrativos.

Essa integração direta elimina a necessidade de exportações e importações manuais entre sistemas distintos, promovendo um fluxo de trabalho contínuo e automatizado. Além de aumentar a eficiência do processo, a solução proposta contribui para reduzir erros causados por inconsistências entre plataformas, agiliza a tomada de decisão e facilita eventuais alterações nos cronogramas. O sistema proposto reformula o fluxo original, integrando o algoritmo genético ao sistema interno, com as seguintes etapas:

1. **Extração e Preparação de Dados:** O módulo do algoritmo genético acessa os dados cadastrados no sistema interno, como informações sobre docentes, turmas, subturmas, componentes curriculares, espaços físicos e restrições, enviados por uma chamada de *API*. Esses dados posteriormente são organizados e validados para garantir que as restrições institucionais, como a limitação de dois turnos por dia para professores, sejam adequadamente consideradas antes do processamento.
2. **Geração de Horários:** O algoritmo genético processa os dados organizados para criar horários otimizados, alocando aulas de forma a respeitar as restrições institucionais. O processo utiliza técnicas de inicialização, avaliação de qualidade, cruzamento, mutação e seleção para explorar soluções viáveis, penalizando violações como conflitos de professores, turmas, salas ou alocações em turnos excessivos, conforme descrito nas subseções 4.1.1 e 4.1.3.
3. **Validação na Interface Administrativa:** Os horários gerados são exibidos na interface administrativa do sistema, onde os gestores analisam os resultados, verificando a conformidade com as restrições institucionais. Na tela é possível visualizar eventuais conflitos, como um professor dando aulas para turmas diferentes no

mesmo horário, permitindo ajustes manuais, se necessário. Essa etapa garante que apenas horários melhor estruturados sejam encaminhados para a próxima fase.

4. **Inserção no Banco de Dados do MRBS:** Após a avaliação, os horários aprovados são inseridos diretamente no banco de dados do MRBS por meio de uma conexão automática, atualizando a ocupação de salas sem a necessidade de exportação de arquivos ou cadastro manual, assim eliminando limitações do fluxo original, como a dependência de intervenções manuais.

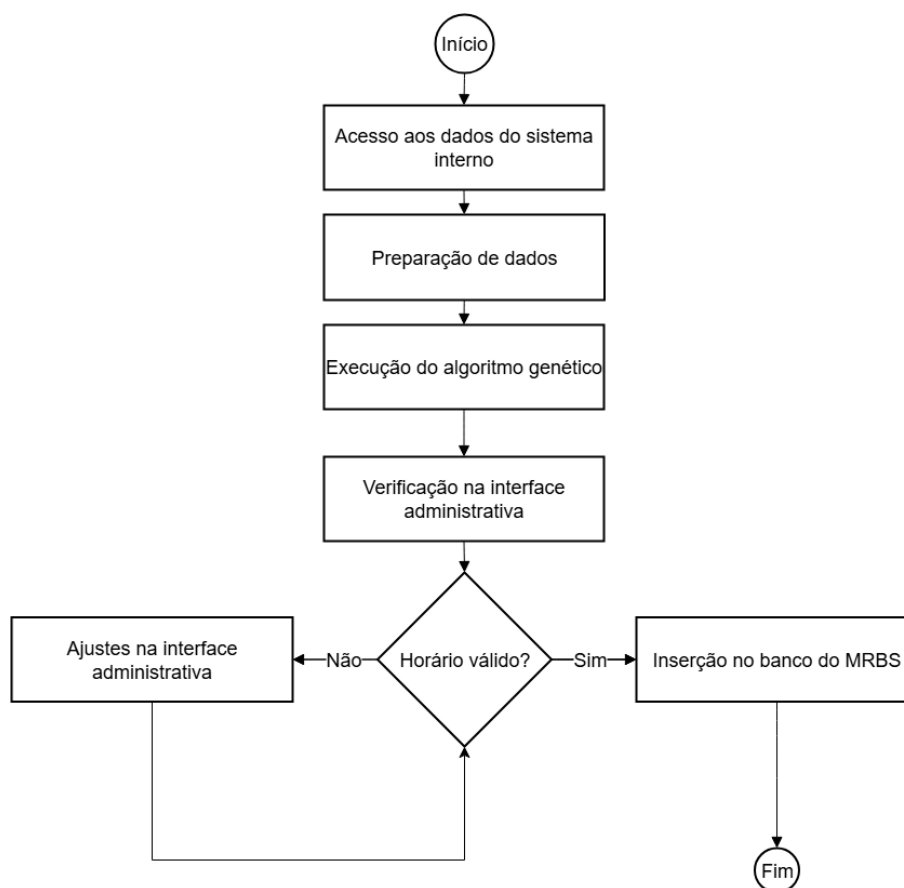


Figura 9. Fluxo de processos da Integração proposta.

A Figura 1 apresentou o fluxo de processos anterior à implementação do sistema proposto, no qual era necessário exportar manualmente os dados do sistema interno de cadastro de disciplinas para o software TimeTables, e posteriormente, deste para o sistema MRBS.

Já a Figura 9 representa o funcionamento após os desenvolvimentos atuais. O novo modelo elimina a necessidade de exportações e importações manuais de arquivos, substituindo-as por uma comunicação direta entre os sistemas por meio de *APIs*. Dessa forma, o algoritmo genético acessa automaticamente os dados do sistema de cadastro de disciplinas, processa as informações e retorna os resultados diretamente para a interface administrativa do sistema, sem a utilização de arquivos externos.

A integração proposta, portanto, substitui o fluxo tradicional baseado em trocas de arquivos por um processo contínuo e automatizado, aumentando a confiabilidade e

reduzindo os gastos operacionais necessários para consolidação dos horários, assim como os gastos que ocorriam com o uso do aSc TimeTables.

4.3. Apresentação da Interface

Com o objetivo de validar a disposição dos elementos e garantir uma experiência de uso intuitiva, foi desenvolvido um conjunto de protótipos para o *front-end* do sistema de agendamento de turmas. A interface foi concebida para permitir tanto a edição e validação administrativa dos horários quanto a visualização pública e filtrada dos cronogramas consolidados. O processo de desenvolvimento seguiu uma abordagem incremental: partindo de uma versão simples em HTML e *JavaScript*, evoluiu para uma aplicação mais completa, integrada ao back-end e ao algoritmo genético implementado no sistema.

4.3.1. Protótipo em HTML/JS

O primeiro passo consistiu em criar um protótipo base em HTML e *JavaScript* puro, com o objetivo de testar a disposição dos elementos visuais e a lógica de marcação dos períodos. O protótipo, apresentado na Figura 10, materializa uma grade semanal com as turmas nas linhas e os dias/turnos/períodos nas colunas. As colunas são agrupadas por dia (segunda a sábado) e subdivididas em 12 períodos (quatro na manhã, quatro na tarde e quatro à noite).



Figura 10. Protótipo da Apresentação dos resultados.

Uma legenda cromática diferenciava os turnos e indicava os estados de cada célula: “disponível” (branco) ou “ocupado” (vermelho). As células respondiam a eventos de clique, permitindo simular a alocação e remoção de aulas, de forma que o usuário pudesse interagir diretamente com a grade para validar sua ergonomia.

O protótipo também definia o esquema de horários e períodos, organizando os dados em listas de turmas, disciplinas e professores. Essa estrutura serviu como base conceitual para as versões seguintes, garantindo uma correspondência direta entre os dados exibidos e as entidades do sistema acadêmico.

4.3.2. Interface de Edição e Validação

Após uma melhor análise do protótipo inicial, a aplicação evoluiu para uma interface mais completa, implementada em HTML e *JavaScript* modularizado, com uso de *Tailwind* CSS para o estilo e *scripts* separados para manipulação de dados e renderização dinâmica. Essa interface passou a se conectar com os dados gerados pelo algoritmo genético (GA), carregando dados vindos diretamente do GA para preencher a grade com os horários propostos pelo sistema.

A tela de edição de horários foi desenvolvida para ser utilizada por coordenadores e administradores, responsáveis por revisar, ajustar e consolidar os horários gerados automaticamente. Ela busca combinar automação e flexibilidade manual, permitindo correções finas antes da exportação para o sistema de reservas de salas (MRBS). Suas principais funcionalidades incluem:

- **Painel de controle superior:** Conjunto de botões para ações principais, como adicionar novas aulas, editar aulas existentes, salvar o horário atual, recarregar dados vindos do Algoritmo Genético (AG) e compartilhar o resultado, o qual gera um link compartilhável do horário. Inclui também controles de *zoom* com indicador percentual e restabelecimento rápido, além de campos de data para envio do cronograma consolidado diretamente ao MRBS.
- **Grade interativa de alocação:** Estrutura principal da interface, onde cada célula representa um período de aula. É possível alocar, mover ou remover aulas por meio de interações diretas com os blocos. As aulas possuem duração variável e podem ocupar múltiplos períodos consecutivos, sendo unificadas visualmente em um único bloco.
- **Verificação automática de conflitos:** Durante a edição, o sistema identifica sobreposições de professores ou salas em tempo real, destacando visualmente os blocos conflitantes com cores distintas e informando exatamente qual professor ou sala está envolvido no conflito.
- **Painel inferior de “Aulas não alocadas”:** Local fixo na parte inferior da tela que exibe os cartões de aulas ainda não posicionadas na grade. Esse painel possui rolagem interna e altura limitada, garantindo que o restante da interface permaneça visível mesmo com grande quantidade de aulas pendentes.
- **Janela de edição e criação de aulas:** Ao editar ou adicionar aulas, o usuário pode definir a turma, matéria, professores (com opção de criar novos docentes no próprio fluxo), sala e duração da aula. Também é possível criar novas matérias e associá-las diretamente a uma turma específica. Um botão flutuante (“Editar aula”) permanece acessível mesmo durante a rolagem, permitindo que o usuário edite rapidamente a aula selecionada sem precisar retornar ao topo da página.

O comportamento da grade é orientado a blocos: cada célula pode receber uma aula com duração variável, e blocos adjacentes se unem automaticamente para representar

disciplinas contínuas. As verificações automáticas e a integração entre o painel de aulas não alocadas e a grade principal visam garantir uma experiência de edição fluida, segura e consistente.

4.3.3. Interface de Visualização e Filtros

A segunda interface do sistema tem caráter consultivo e é destinada à visualização pública ou interna dos horários consolidados, sendo voltada a alunos, docentes e demais interessados. Diferentemente da tela de edição, ela não permite alterações manuais: seu objetivo é apresentar o cronograma de forma organizada e acessível.

Essa interface é aberta a partir de um link gerado na tela de edição, por meio do botão “Compartilhar”. Ao acionar esse recurso, o sistema cria um endereço único contendo o identificador do arquivo salvo (por exemplo, `vizualizar.html?file=2026220251019205706.json`). Qualquer usuário com o link pode visualizar o cronograma correspondente diretamente, sem necessidade de autenticação ou acesso ao ambiente de edição. O cronograma é então carregado automaticamente a partir do servidor e renderizado em uma grade estática.

Entre suas principais funcionalidades estão:

- **Zoom interativo:** Permite redimensionar a grade facilitando tanto a visualização geral quanto a análise de detalhes de cada aula.
- **Filtros dinâmicos:** Turma: exhibe apenas os horários de uma turma específica; Professor: mostra todas as aulas ministradas por um docente; Sala: apresenta as aulas realizadas em um ambiente específico. Cada filtro reorganiza a exibição da grade, buscando manter a consistência visual e evidenciar as informações relevantes para o usuário.
- **Cabeçalhos e modos de visualização:** No modo completo, são exibidas todas as turmas simultaneamente (linhas representam turmas e colunas os dias da semana). Nos modos filtrados (por turma, professor ou sala), as linhas passam a representar os dias da semana, mantendo os períodos como colunas, de forma a destacar os horários ocupados.
- **Identificação contextual:** O nome do cronograma carregado é exibido logo abaixo do título principal, informando ao usuário qual horário está sendo visualizado, visando facilitar o reconhecimento de versões diferentes de cronogramas salvos.

O filtro de Professor e Sala introduz um mecanismo de “*lanes*” (faixas paralelas) para representar graficamente aulas simultâneas: quando um mesmo professor ministra duas aulas no mesmo período, ou uma sala é utilizada em mais de uma turma simultaneamente, o sistema cria faixas sobrepostas para evitar colisões visuais. Essa solução busca manter a legibilidade da grade e permitir uma análise clara de disponibilidade, possibilitando identificar sobrecargas e conflitos de forma imediata.

Essa abordagem visa a visualização acessível, autoexplicativa e compartilhável, integrando a etapa final do fluxo do sistema — desde a geração do horário pelo Algoritmo Genético, passando pela edição manual, até sua divulgação pública via link direto.

4.3.4. Integração dos Módulos e Encaminhamento

Os módulos desenvolvidos (edição e visualização) compartilham uma mesma base de dados e utilizam o adaptador *AGADAPTER*, responsável por converter o formato de saída do algoritmo genético para o modelo visual esperado pelos editores e visualizadores.

Essa integração buscar garantir consistência entre as etapas e permitir que todo o fluxo ocorra dentro de um único ecossistema web. A validação final dos horários é feita visualmente na tela de edição, e, após a consolidação, o cronograma pode ser carregado na tela de visualização para análise pública ou institucional.

Com essas implementações, o sistema visa atingir o objetivo de integrar o algoritmo de geração automática com uma interface visual acessível, responsiva e coerente com as necessidades administrativas do campus.

4.4. Exportação de dados para o MRBS

O processo de exportação de dados para o sistema MRBS (*Meeting Room Booking System*) representa a etapa final da integração desenvolvida neste trabalho, consolidando o cronograma validado na interface administrativa e registrando-o diretamente no banco de dados institucional. Essa integração elimina a necessidade de exportações intermediárias de arquivos XML ou semelhantes, substituindo-as por um envio direto via *API*, garantindo a sincronização automática dos horários e o uso imediato das salas alocadas no sistema de reservas.

A partir do momento em que o horário é validado na interface de edição, o usuário pode realizar a consolidação definitiva, enviando todos os dados das alocações das salas para o MRBS. O sistema se encarrega de interpretar as informações e gerar os registros de reservas correspondentes, respeitando as configurações de duração de cada aula.

4.4.1. Integração com o banco de dados e controle de períodos

A comunicação com o MRBS ocorre por meio de uma estrutura de requisições automatizadas que interagem diretamente com o banco de dados do sistema. Cada registro de aula validado na tela de edição é convertido em uma entrada correspondente no formato aceito pelo MRBS, associando-se à sala, à data, ao horário de início e à duração correspondentes.

Durante a elaboração do módulo, foi necessário considerar as particularidades do calendário acadêmico e das configurações do câmpus, especialmente no que diz respeito aos turnos noturnos. As turmas com quatro períodos letivos noturnos, por exemplo, têm suas aulas iniciadas às 18h30, enquanto aquelas com menos de quatro períodos iniciam às 19h00, como pode ser visto na Figura 11. Essa lógica foi incorporada ao processo de exportação para garantir que os registros inseridos no MRBS reflitam com precisão a ocupação real dos ambientes, evitando sobreposições e divergências entre horários teóricos e práticos.

O sistema também verifica automaticamente a duração total de cada aula antes da exportação, ajustando os intervalos de acordo com o número de períodos associados. Essa checagem garante que as reservas no MRBS correspondam exatamente à quantidade de blocos de tempo utilizados por cada turma, respeitando a duração configurada no

cronograma acadêmico.

Como a maioria das salas de aula no MRBS utiliza 8 períodos noturnos, gerando incompatibilidade tanto com o sistema de cadastro de disciplinas quanto com a interface desenvolvida que possuem 4, uma tradução dos períodos foi realizada, sendo assim, o 4º período noturno na interface (das 21:30 às 22:10), corresponde aproximadamente ao 7º período noturno do MRBS (das 21:30 às 22:00). Assim ocorre com os outros horários noturnos, em que traduções entre a interface e o sistema de alocação de salas foram realizadas para tentar gerar uma maior coerência na visualização.

Período	seg 10 de nov.	ter 11 de nov.	qua 12 de nov.	qui 13 de nov.	sex 14 de nov.
08:00 às 08:55	SUP. Ciência da Computação 1 - Introdução ao Desenvolvimento Web	SUP. Ciência da Computação 1 - Eletrônica Digital	SUP. Ciência da Computação 1 - Introdução à Programação	SUP. Ciência da Computação 1 - Eletrônica Digital	SUP. Ciência da Computação 1 - Computação e Sociedade
08:55 às 09:50					SUP. Ciência da Computação 1 - Introdução à Programação
10:10 às 11:05					
11:05 às 12:00					
12:00 às 13:30 (Intervalo)					
13:30 às 14:25	Tec. Informática para Internet 1 - Sistemas e Conectividade	Tec. Informática para Internet 1 - Introdução à Programação	Tec. Informática para Internet 1 - Comunicação e Expressão	Tec. Informática para Internet 1 - Banco de Dados	
14:25 às 15:20					
15:40 às 16:35					
16:35 às 17:30					
17:30 às 18:30 (Intervalo)					
18:30 às 19:00					TEC. em Eletromecânica 1 - Informática
19:00 às 19:30	Tec. Desenv. de Sistemas 2 - Desenvolvimento de Sistemas	Tec. Desenv. de Sistemas 2 - Rodízio Desenv. Sistemas			
19:30 às 20:00					
20:00 às 20:30					
20:30 às 21:00					
21:00 às 21:30					
21:30 às 22:00					
22:00 às 22:30					

Figura 11. Visualização semanal no sistema MRBS após exportação automática dos horários.

Tec Informática para Internet 1 - Introdução à Programação

Event details

Início: 13:30 às 14:25, terça-feira, 11 de novembro de 2025
Duração: 4 períodos
Fim: 14:25 às 15:20, terça-feira, 11 de novembro de 2025
Sala: Lab. Informática - Laboratório 116
Última Atualização: quinta-feira, 6 de novembro de 2025 16:26
Marcado por: admin
Modified by:
Tipo: Interno
Descrição: Curso: Tec Informática para Internet
Fase: 1
Prof(s): João Augusto da Silva Bueno
Estado da reserva: Aprovado
Estado da reserva: Confirmado
Repetir Tipo: Nenhum

[Voltar à página anterior](#)

Figura 12. Detalhes de uma alocação no MRBS.

A interface do MRBS oferece visualizações diárias, semanais e mensais, o que facilita o acompanhamento e a manutenção das reservas. As informações exportadas

pela nova interface incluem não apenas o nome da turma e da disciplina, mas também o curso de origem e os professores na parte da descrição referente a alocação específica, permitindo que os setores administrativos possuam variados dados sobre aquela reserva, como é possível identificar na Figura 12.

5. Resultados

Esta seção apresenta os resultados obtidos com a implementação do sistema de geração e visualização de horários, contemplando as interfaces desenvolvidas e o funcionamento geral das etapas de edição, validação e consulta. As Figuras 13, 14, 15, 16 e 17 ilustram o comportamento do sistema após a integração completa entre o algoritmo genético e o *front-end Web* construído.

O sistema atinge seu objetivo principal ao permitir que os horários gerados automaticamente pelo algoritmo sejam visualizados, revisados e consolidados por meio de uma interface intuitiva e responsiva. A interface final foi projetada de modo a atender tanto usuários administrativos, responsáveis pela validação das alocações, quanto usuários de consulta, que necessitam apenas visualizar os cronogramas consolidados.



Figura 13. Tela de Apresentação de resultados.

A Figura 13 apresenta a tela de edição e validação de horários, destinada aos coordenadores de curso e gestores acadêmicos. Nessa interface, as turmas são organizadas em linhas e os períodos em colunas, permitindo ao usuário identificar visualmente as aulas já alocadas, os espaços disponíveis e possíveis conflitos de professores ou salas, que são destacados automaticamente em vermelho e roxo.

No painel inferior fixo, são exibidas as aulas não alocadas, apresentadas em forma de cartões, que podem ser alocadas manualmente na grade conforme a disponibilidade de tempo e sala. Esse painel possui rolagem interna e mantém-se visível durante toda a edição, facilitando a reorganização dinâmica das aulas.

A interface inclui um painel superior de controle, que reúne as principais ações do usuário: adição e edição de aulas, salvamento, recarregamento dos dados provenientes do algoritmo genético e envio direto do cronograma consolidado ao sistema MRBS.

A interface também oferece controles de zoom com indicador percentual, permitindo ajustar a visualização da grade sem perda de legibilidade, além de mecanismos automáticos de validação que verificam, em tempo real, conflitos de professor e sala durante a edição.

Por fim, ao posicionar o cursor sobre um *card* de aula, o sistema exibe informações detalhadas, como o professor responsável, a duração da aula, a sala e a turma correspondente, recurso presente em todas as telas do sistema e que auxilia na inspeção rápida dos dados.

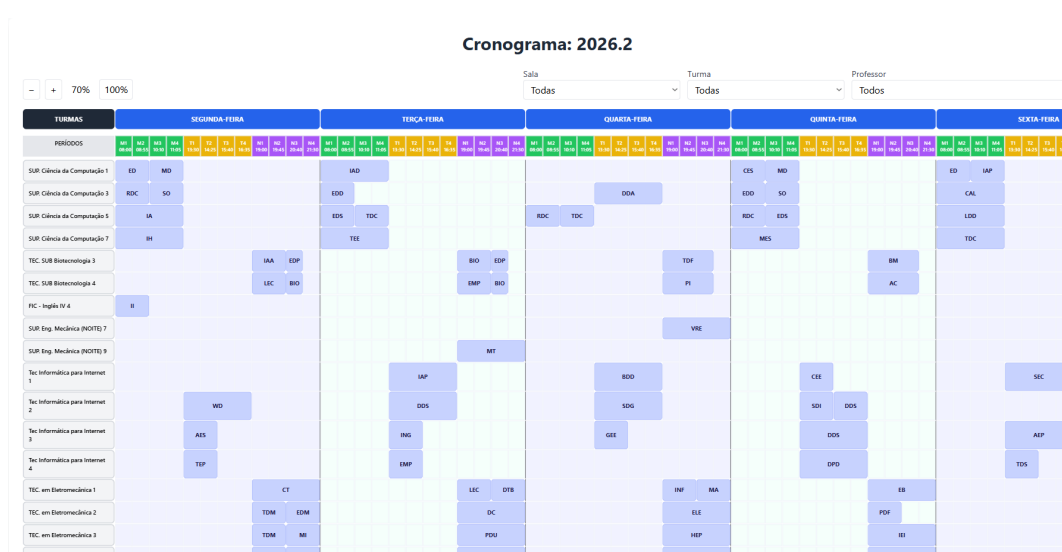


Figura 14. Tela de Visualização.

A Figura 14 mostra a tela de visualização geral, que apresenta a disposição completa das turmas e disciplinas para um semestre específico. Essa visão panorâmica permite avaliar o equilíbrio global do cronograma, permitindo identificar sobrecargas ou períodos ociosos. Além disso, o sistema exibe o nome do horário atualmente aberto, garantindo clareza sobre qual horário está sendo visualizado. A legenda de cores diferencia os turnos e a interface mantém sua responsividade possibilitando o uso do zoom para ajuste de visualização e agora aplicação de filtros dinâmicos.

A Figura 15 destaca o modo de visualização por turma, obtido a partir da aplicação dos filtros disponíveis na interface. Nessa configuração, a grade é reorganizada de forma que as linhas representem os dias da semana e as colunas representem os períodos diários, onde cada *card* de aula contém uma *badge* indicando qual professor leciona a aula e em qual sala será ministrada, permitindo uma leitura individual e detalhada da rotina semanal de uma turma específica. Essa visualização é especialmente útil para docentes e estudantes que desejam consultar o horário completo de uma única turma.

A Figura 16 apresenta o modo de visualização por professor, que exibe todas as disciplinas atribuídas a um mesmo docente em diferentes cursos e turmas. Esse modo im-

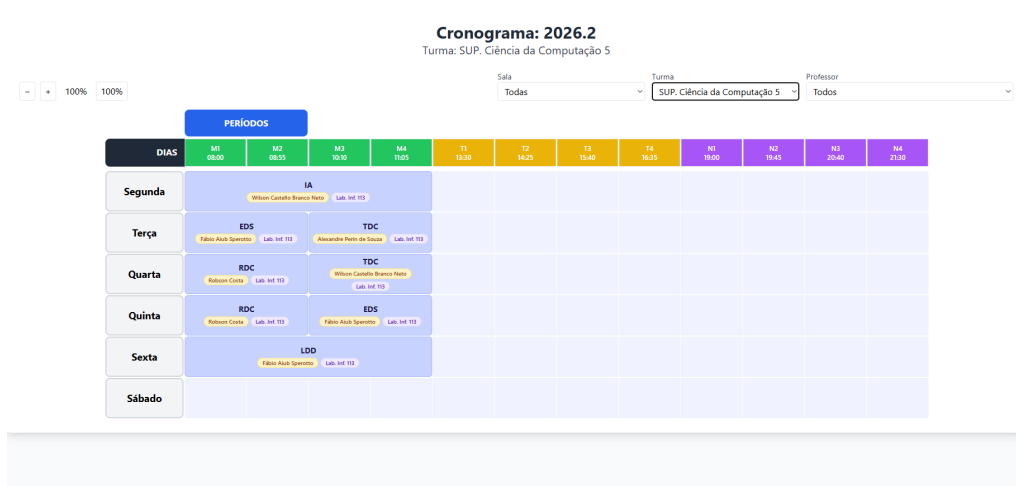


Figura 15. Tela de Visualização com filtro de Turma.

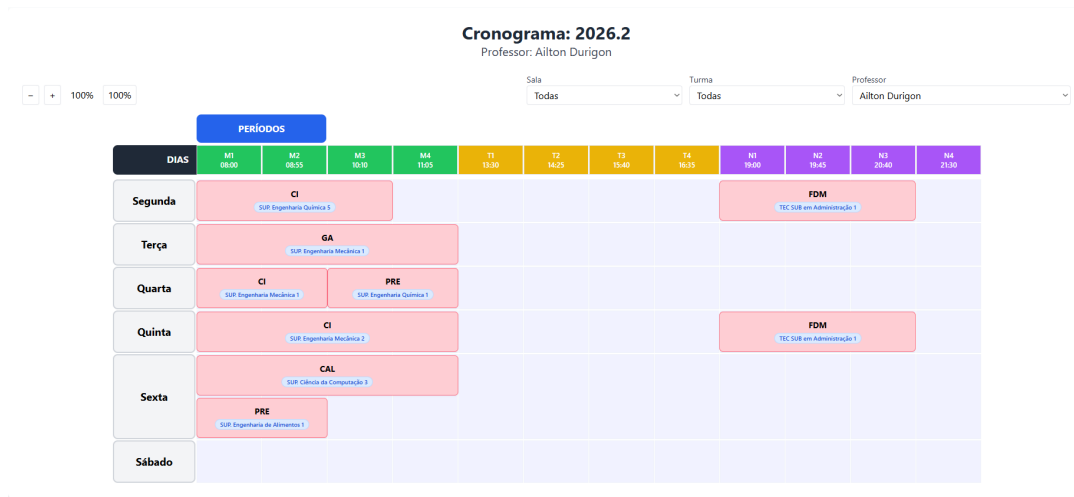


Figura 16. Tela de Visualização com filtro de Professor.

plementa o recurso de faixas paralelas, as quais são renderizadas automaticamente quando o sistema detecta aulas simultâneas para o mesmo professor. Além disso, os *cards* de aula exibem claramente a turma à qual cada disciplina pertence, permitindo identificar rapidamente a origem de cada alocação e compreender a distribuição das aulas entre cursos diferentes. Essa representação busca facilitar a identificação de conflitos de horário e auxilia na redistribuição de aulas durante a etapa de validação.

A Figura 17 apresenta o modo de visualização por sala, que exibe todas as aulas alocadas em um mesmo ambiente físico ao longo da semana. Esse modo permite acompanhar de forma clara a utilização de cada sala, identificando quais turmas e disciplinas ocupam o espaço em cada período. Assim como na visualização por professor, o sistema implementa o recurso de faixas paralelas, renderizadas automaticamente quando são detectadas aulas simultâneas na mesma sala, garantindo que todos os eventos sejam representados sem sobreposição visual.

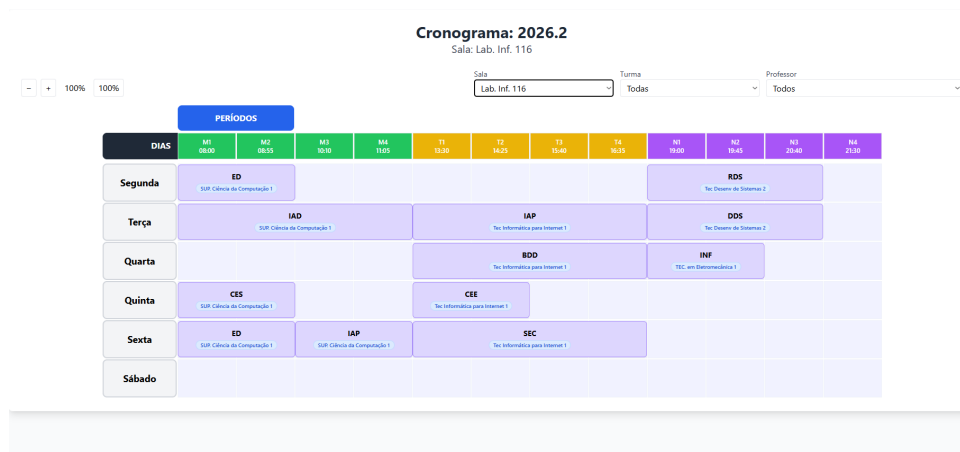


Figura 17. Tela de Visualização com filtro de Sala.

The dialog box 'Adicionar aulas (vão para "Não alocadas")' is used for adding new classes. It features several input fields: 'Turma' (SUP. Ciência da Computação 1), 'Matéria' (Introdução ao Desenvolvimento Web), 'Professor(es)' (a list of names including Edinilson da Silva Vida, Leonardo Bravo Estacio, Jeferson Fraytag, Juliana Mercedes Rheinheimer, Wilson Castello Branco Neto, and Robson Costa), 'Sala (opcional)' (sem sala), 'Duração (períodos)' (1), and 'Quantidade de cards' (1). The 'Adicionar' button is highlighted in green.

Figura 18. Tela de Adicionar nova aula.

A Figura 18 apresenta a interface de adição de aulas, acessível diretamente a partir do painel superior da tela de edição. Essa funcionalidade permite que o coordenador insira manualmente novas aulas na grade, ampliando ou ajustando o cronograma gerado pelo algoritmo genético. O formulário oferece campos para seleção da turma, disciplina, professor(es), sala e duração, além da opção de criar novos registros de disciplinas ou docentes, caso ainda não existam no sistema.

As aulas adicionadas são inicialmente direcionadas ao painel de aulas não alocadas, permanecendo pendentes até que sejam posicionadas na grade principal. Esse fluxo assegura que novas inserções não gerem conflitos imediatos e possibilita ao usuário realizar ajustes finos de alocação posteriormente.

As telas buscam a efetividade da integração do processo automatizado de geração de horários com um ambiente visual de gestão acadêmica, possibilitando tanto a análise geral quanto a consulta específica por filtros. Além disso, a estrutura modular visa permitir a adaptação do sistema para diferentes semestres, pois possibilita o salvamento de múltiplos arquivos, um para cada semestre, por exemplo, garantindo sua reutilização para a instituição.

6. Conclusões

Este trabalho apresentou o desenvolvimento de uma solução integrada para o problema de *timetabling* no IFSC Câmpus Lages, com foco na automação do processo de geração e validação de horários acadêmicos. A proposta teve como base a aplicação de um algoritmo genético para a criação de cronogramas otimizados, associado a uma arquitetura de integração entre o sistema interno de cadastro de disciplinas e o ambiente de reserva de salas MRBS.

Os objetivos definidos inicialmente foram atendidos. Primeiramente, foi realizada a análise do fluxo de geração de horários vigente no campus, identificando as limitações do processo tradicional baseado na exportação manual de arquivos XML e na utilização de softwares externos, como o TimeTables. Em seguida, foi desenvolvido e ajustado um algoritmo genético em Python capaz de gerar automaticamente combinações de horários que evitam as restrições institucionais, como a disponibilidade de docentes, turmas compartilhadas, períodos e salas.

Posteriormente foi desenvolvida a integração entre o algoritmo e o sistema interno de cadastro de disciplinas, permitindo a troca automática de dados por meio de APIs. Essa integração eliminou a necessidade de exportações intermediárias, reduzindo o tempo de processamento e a dependência de intervenção manual. Além disso, o sistema passou a permitir a consolidação direta dos horários no banco de dados do MRBS, simplificando o registro de reservas e a atualização da ocupação das salas.

Foram também criadas duas interfaces *Web* complementares. A primeira, voltada à edição e validação administrativa, possibilita ajustes manuais, detecção automática de conflitos e exportação direta para o MRBS. A segunda, destinada à visualização e consulta pública, apresenta filtros por professor, turma e curso, e modos de exibição diferenciados.

Como resultado, o novo sistema consolidou um fluxo contínuo e automatizado que substitui o modelo anterior baseado em trocas de arquivos, aumentando a confiabilidade do processo e integrando todas as etapas da geração de horários — desde o processamento inicial no algoritmo genético até o registro final no MRBS.

Apesar dos avanços alcançados, algumas atividades podem ser exploradas como trabalhos futuros. Uma primeira possibilidade consiste na realização de testes e validações junto aos usuários finais, especialmente coordenadores de curso e setores administrativos, visando à coleta de *feedback* sobre a usabilidade, clareza das informações apresentadas e eficiência operacional da solução.

Além disso, propõe-se o aprofundamento da análise do comportamento do algoritmo genético por meio da documentação sistemática de observações técnicas já identificadas durante o desenvolvimento, como padrões recorrentes de conflitos, impacto da ordenação inicial das aulas e influência da duração das disciplinas na viabilidade das

soluções geradas.

Também se sugere o estudo da variação dos parâmetros do algoritmo genético — tais como taxa de mutação, tamanho da população e número de gerações — a fim de avaliar o impacto dessas configurações tanto na qualidade das soluções quanto no tempo de execução, permitindo o refinamento progressivo do modelo adotado.

A solução foi desenvolvida de forma específica para os sistemas e regras acadêmicas do IFSC Câmpus Lages, considerando suas particularidades de cadastro de disciplinas, disponibilidade docente e integração com sistemas institucionais existentes.

Ainda assim, a abordagem adotada permite que, mediante retrabalho e adequações técnicas, a solução possa ser adaptada para outras instituições de ensino que enfrentam problemas semelhantes na organização de horários acadêmicos. Esse processo de adaptação envolveria ajustes na modelagem dos dados, nas regras de negócio e nas integrações com sistemas locais, caracterizando a solução como uma base reutilizável, porém não imediatamente transferível sem customizações.

Em síntese, o trabalho alcançou seus objetivos ao propor e implementar uma arquitetura integrada e automatizada para a geração de horários acadêmicos, contribuindo para a modernização dos processos administrativos e a redução de esforços manuais.

Referências

- Albino, G. C. (2022). Algoritmos genéticos aplicados ao problema de alocação de horários e professores em escolas do ensino fundamental. Accepted: 2022-12-23T21:16:54Z Publisher: Florianópolis, SC.
- aSc Applied Software (2025). asctimetables. Acesso em: 26 maio 2025. Disponível em: <https://www.asctimetables.com/>.
- Bianchini, I. R. e Silva, O. J. H. d. (2020). Algoritmo Genético distribuído para problema de Timetabling. Accepted: 2022-03-24T19:49:38Z.
- Capaverde, A. M. (2025). UNIVERSIDADE FEDERAL DE SANTA CATARINA CENTRO TECNOLÓGICO - CTC SISTEMAS DE INFORMAÇÃO.
- Chikwiriro, H., Pharaoh, C., M, M. P., e David, N. (2014). Use of genetic algorithms in rigorous university timetabling. *International Journal of Engineering Research & Technology (IJERT)*, 3(3).
- Davenport, T. H. (1998). Putting the enterprise into the enterprise system.
- De Lima, A., Carvalho, S., Dias, D. M., Naredo, E., Sullivan, J. P., e Ryan, C. (2022). GRAPE: Grammatical Algorithms in Python for Evolution. *Signals*, 3(3):642–663.
- Eiben, A. e Smith, J. (2015). *Introduction to Evolutionary Computing*. Natural Computing Series. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., e Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press.
- Kim, J. e Yoo, S. (2019). Software review: DEAP (Distributed Evolutionary Algorithm in Python) library. *Genetic Programming and Evolvable Machines*, 20(1):139–142.
- Liden, R. (2008). *Algoritmos Genéticos*. Brasport, Rio de Janeiro, 2 edition.

- Luftman, J. (2004). Assessing business-IT alignment maturity. In *Strategies for Information Technology Governance*, pages 99–128. IGI Global Scientific Publishing.
- MRBS Team (2025). MRBS: Introduction. Acesso em: 26 maio 2025. Disponível em: <https://sourceforge.net/projects/mrbs/>.
- Rebonatto, M. T., Fabian, M. F., e Foschiera, F. G. (2025). Um algoritmo genético para a problema de geração de horários a partir de currículos com disciplinas compartilhadas. 19:e025016–e025016.
- Sani, H. M. e Yabo, M. M. (2016). Solving timetabling problems using genetic algorithm technique. *International Journal of Computer Applications*, 134(15):33–38.
- Santos, J. L. d. L. (2016). Uma ferramenta para geração de grades horárias utilizando algoritmos genéticos. Accepted: 2018-03-20T23:36:18Z Publisher: Universidade Federal da Paraíba.
- Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127.
- Thakare, S., Nikam, T., e Patil, M. (2020). Automated timetable generation using genetic algorithm. *International Journal of Engineering Research and*, V9.
- Vasconcelos, A. Y. S. d. (2021). Algoritmos genéticos para a resolução de problemas de alocação de horário de aulas na UFGD. Accepted: 2021-11-19T14:33:16Z Publisher: Universidade Federal da Grande Dourados.
- Xu, H. e Petrie, K. (2012). Self-Learning Genetic Algorithm For Constrains Satisfaction Problems. In Jones, A. V., editor, *2012 Imperial College Computing Student Workshop*, volume 28 of *Open Access Series in Informatics (OASICs)*, pages 156–162, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISSN: 2190-6807.
- Zhang, Y., Ogura, H., ma, X., Kuroiwa, J., e Odaka, T. (2014). A genetic algorithm using infeasible solutions for constrained optimization problems. *The Open Cybernetics & Systemics Journal*, 8:904–912.