SmarTask - Bigger Companies, Better Scheduling

Rafael Kauati, Gabriel Teixeira, Vitalie Bologa, João Monteiro
Orientadores: Amaro de Sousa, Luís Seabra Lopes, José Moreira

Projeto em Engenharia Informática, 3º ano, LEI.

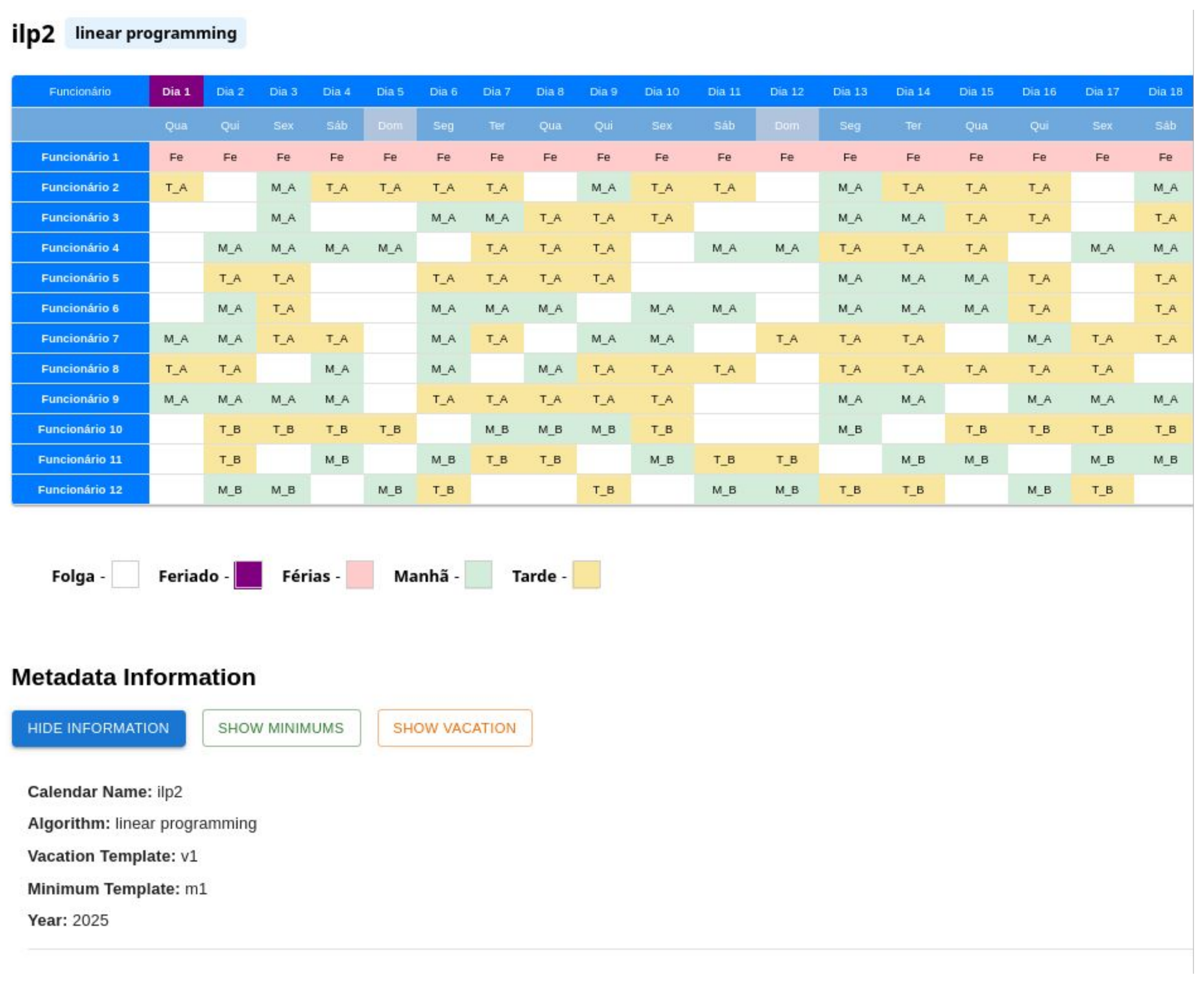
2025

https://pi-smarttask.github.io/website/

Abstract

SmarTask is a project and application focused on testing work schedule generation, allowing the configuration of various input data and optimization algorithms. Among the main features:

- Defining vacations per employee (vacation template);
- Configuring minimums per shift and team (minimum coverage template);
- Recording employee information and preferences (teams they can work in and priority);
- Integration with schedule generation algorithms, which classify each day of each employee as a shift, day off or vacation;



ilp2 linear programming

Funcionário	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7	Dia 8	Dia 9	Dia 10	Dia 11	Dia 12	Dia 13	Dia 14	Dia 15	Dia 16	Dia 17	Dia 18
	Qua	Qui	Sex	Sáb	Dom	Seg	Ter	Qua	Qui	Sex	Sáb	Dom	Seg	Ter	Qua	Qui	Sex	Sáb
Funcionário 1	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe	Fe
Funcionário 2	T_A		M_A	T_A	T_A	T_A	T_A		M_A	T_A	T_A		M_A	T_A	T_A	T_A		M_A
Funcionário 3			M_A			M_A	M_A	T_A	T_A	T_A			M_A	M_A	T_A	T_A		T_A
Funcionário 4		M_A	M_A	M_A	M_A		T_A	T_A	T_A		M_A	M_A	T_A	T_A	T_A		M_A	M_A
Funcionário 5		T_A	T_A			T_A	T_A	T_A	T_A				M_A	M_A	M_A	T_A		T_A
Funcionário 6		M_A	T_A			M_A	M_A	M_A		M_A	M_A		M_A	M_A	M_A	T_A		T_A
Funcionário 7	M_A	M_A	T_A	T_A		M_A	T_A		M_A	M_A		T_A	T_A	T_A		M_A	T_A	T_A
Funcionário 8	T_A	T_A		M_A		M_A		M_A	T_A	T_A	T_A		T_A	T_A	T_A	T_A	T_A	
Funcionário 9	M_A	M_A	M_A	M_A		T_A	T_A	T_A	T_A	T_A			M_A	M_A		M_A	M_A	M_A
Funcionário 10		T_B	T_B	T_B	T_B		M_B	M_B	M_B	T_B			M_B		T_B	T_B	T_B	T_B
Funcionário 11		T_B		M_B		M_B	T_B	T_B		M_B	T_B	T_B		M_B	M_B		M_B	M_B
Funcionário 12		M_B	M_B		M_B	T_B			T_B		M_B	M_B	T_B	T_B		M_B	T_B	

Folga - ⬜ Feriado - 🟪 Férias - 🟥 Manhã - 🟩 Tarde - 🟨

Metadata Information

HIDE INFORMATION | SHOW MINIMUMS | SHOW VACATION

Calendar Name: ilp2
Algorithm: linear programming
Vacation Template: v1
Minimum Template: m1
Year: 2025

Fig 1- Visualization of generated schedule, with its labels and also with the schedule metadata

Architecture

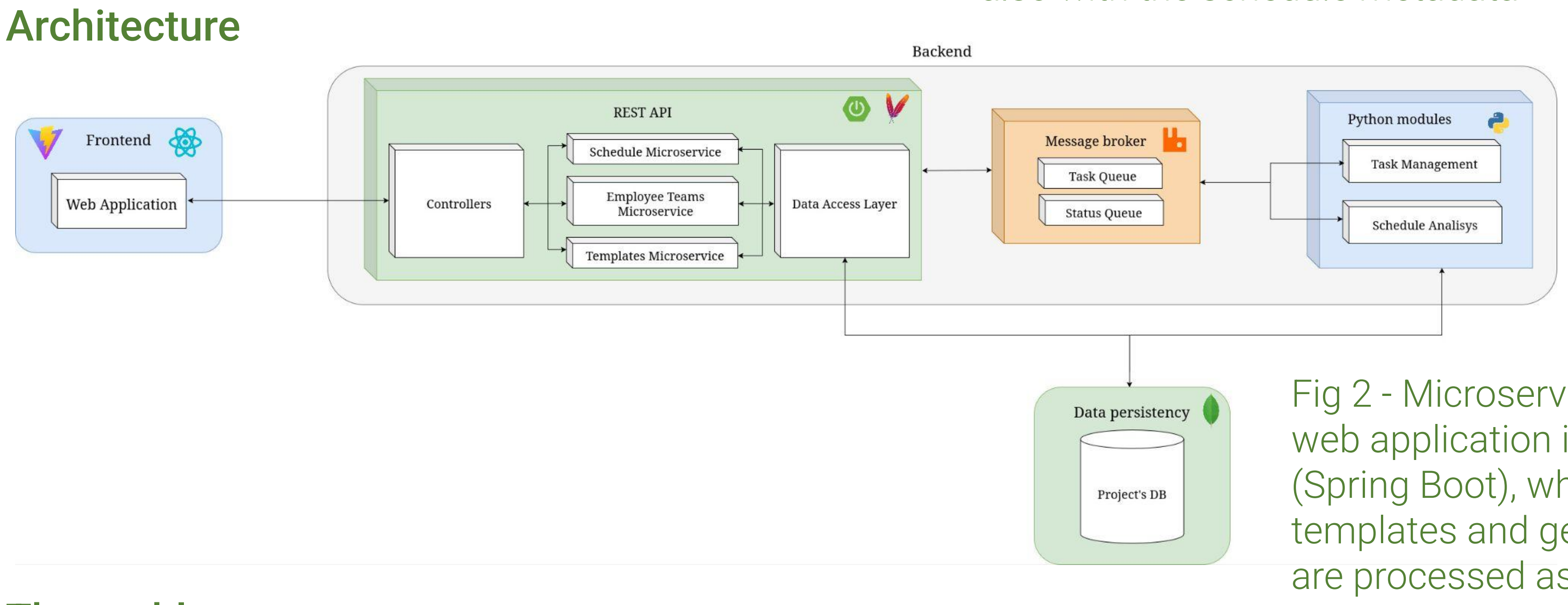


Fig 2 - Microservices-based architecture: the web application interacts with REST APIs (Spring Boot), which coordinate team data, templates and generated calendars. Tasks are processed asynchronously via RabbitMQ by Python modules specialized in managing schedule generation requests (tasks), as well as modules that perform qualitative analysis based on KPIs of previously generated calendars, with persistence in a Mongo database.

The problem

The developed algorithm must generate solution by following a sequence of rules/restrictions, enforced to each employee to the whole 365 days, that are :

- Exactly 223 working days per year per employees;
- Maximum of 22 Sundays/holidays worked;
- Maximum of 5 consecutive working days;
- Prohibition of T→M transitions on consecutive days;
- Imposition of days off on defined vacation days;

Algorithms must also generate schedules that cover minimum employee requirements for each day, shift, and team, in addition to the previously mapped restrictions.

Best solutions

The list below contains the best algorithms, schedule generating algorithms, that were used in the final project :

1. Integer linear programming
2. Hill Climbing
3. Greedy Randomized
4. Greedy Randomized refined with Hill Climbing

Tests and Results

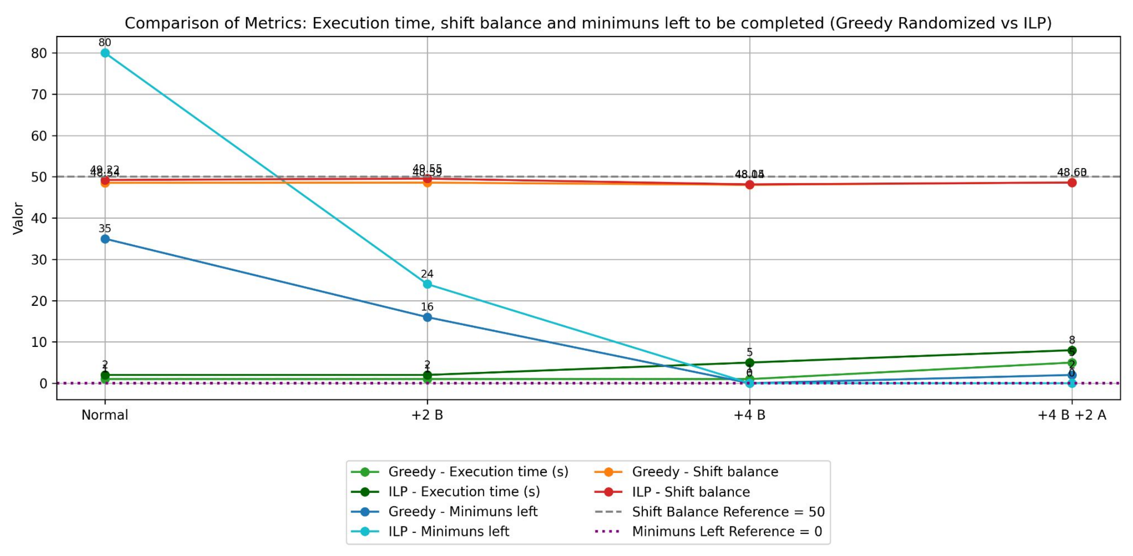


Fig 3 - Test results of the execution time, shift balance and the number of minimums left to be fulfilled from the 2 of the best solutions developed, to 4 different testing configurations of team-employees, a base scenario,12 employees distributed irregularly between two teams (non-ideal), and other scenarios where its added more employees between teams.