

Dados do Projeto

Nome: CGIA (Controle Genérico de Instituições Acadêmicas)

Github Aplicativo: <https://github.com/leodirinaarluccas/CGIA>

Github Backend: <https://github.com/leodirinaarluccas/CGIA-backend>

Product Owner: Pedro Giuliano Farina

Scrum Master: Rafael Galdino Marinho

Development Team:

- Artur Santos de França Carneiro 31724833
- Lucas Fernandez Nicolau 31750923
- Leonardo Shun Mendes de Rosa 31718582
- Pedro Giuliano Farina 31734391
- Rafael Galdino Marinho 31731740

O CGIA (Controle Genérico de Instituições Acadêmicas) é uma plataforma educacional com o intuito de auxiliar escolas de menor porte no gerenciamento de instituição com o foco nas relações de avaliação entre aluno e professor como lançar e acessar notas. O sistema busca oferecer aos alunos e professores uma interface que facilite a realização de tarefas do cotidiano escolar. O projeto busca construir um ambiente seguro e confiável.

Artefatos

- Levantamento de Requisitos
- Historia de Usuários
- Modelo de Diagrama de Classe
- Documentação dos Endpoints da API

Backend: Utilizou-se a framework Vapor na versão 3.0. Esta framework foi criada com o intuito de possibilitar o uso da linguagem Swift para a criação de serviços Web, como APIs.

Frontend: Utilizou-se o iOS como plataforma para a realização do Frontend na linguagem Swift.

Links

Kanban: <https://airtable.com/shr6ViidEhP4QOP1q>

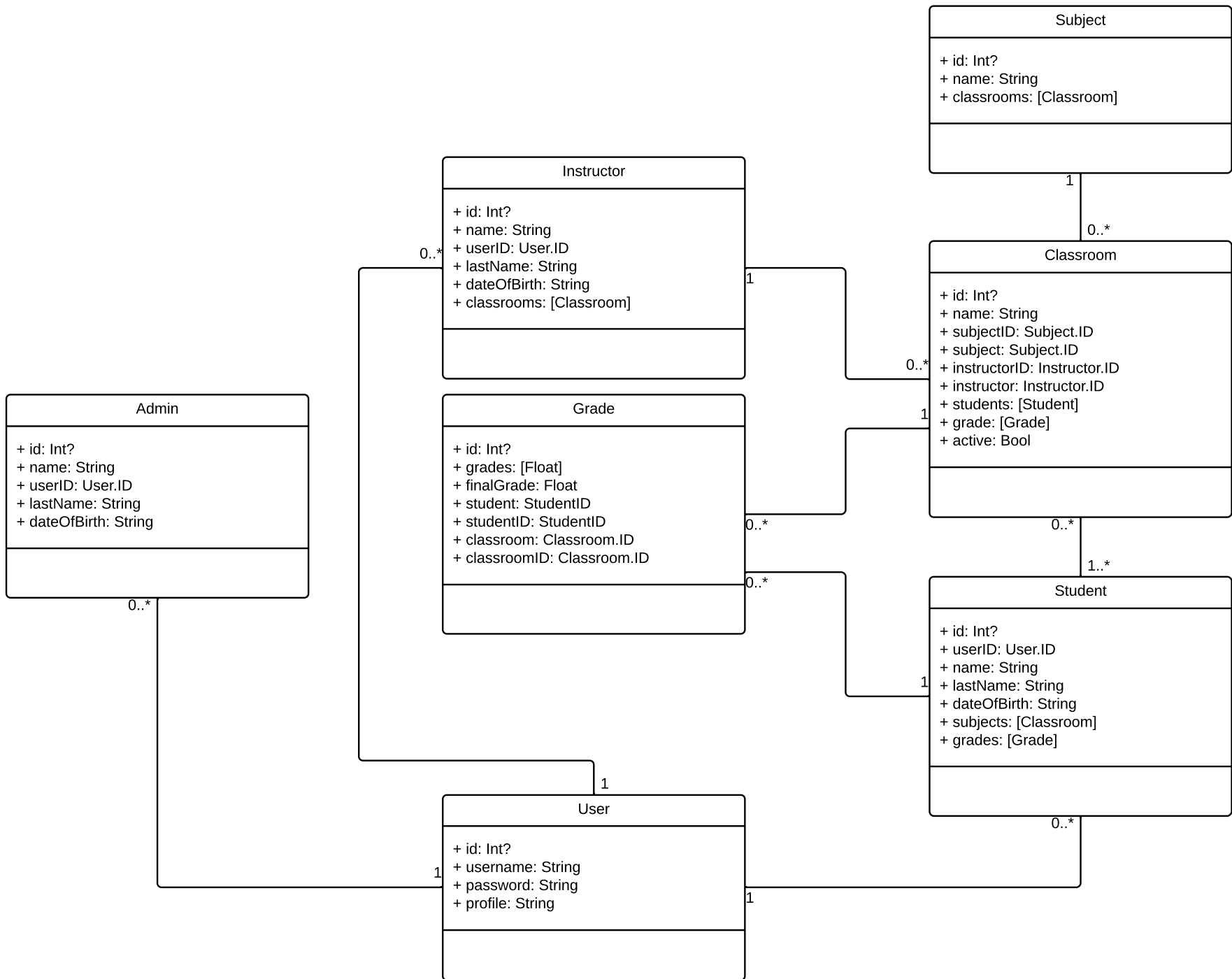
Cronograma: <https://airtable.com/shr4Uc3HjnZEuA1vq>

História de usuário

- Eu como **administrador** quero cadastrar/editar/remover minha instituição para manejar os professores e alunos.
- Eu como **administrador** quero cadastrar/editar/remover meus professores para controlar seu acesso ao sistema.
- Eu como **administrador** quero cadastrar/editar/remover meus alunos para controlar seu acesso ao sistema.
- Eu como **administrador** quero cadastrar/editar/remover matérias no sistema para criar turmas dessas matérias.
- Eu como **administrador** quero cadastrar/editar/remover turmas no sistema para atribuir meus professores e alunos.
- Eu como **administrador** quero cadastrar/editar/remover professores e alunos em turmas para organizar o sistema.
- Eu como **professor** quero ver as matérias que ministro para organizar minha agenda.
- Eu como **professor** quero cadastrar/editar/remover as notas dos alunos para que o sistema compute, calcule as médias e informe os alunos.
- Eu como **professor** quero editar meus dados para manter o sistema atualizado.
- Eu como **aluno** quero me cadastrar numa instituição para fazer parte do sistema.
- Eu como **aluno** quero ver as minhas notas no sistema para me informar.
- Eu como **aluno** quero editar meus dados para manter o sistema atualizado.

Levantamento de Requisitos

REQUISITOS FUNCIONAIS	REQUISITOS NAO-FUNCIONAIS
Cadastrar instituições	Confiabilidade
Editar instituições	Integridade dos dados
Remover instituições	Disponibilidade
Cadastrar professores	Atomicidade
Editar professores	
Remover professores	
Cadastrar alunos	
Editar alunos	
Remover alunos	
Cadastrar turmas	
Editar turmas	
Remover turmas	
Atribuir professores, alunos e disciplinas em turmas	
Editar professores, alunos, e disciplinas em turmas	
Remover professores, alunos e disciplinas em turmas	
Lancar notas dos alunos	
Editar notas dos alunos	
Remover notas dos alunos	
Visualizar notas	
Visualizar disciplinas	
Cadastrar disciplinas	
Editar disicplinas	
Remover disciplinas	



Endpoints	Descrição	Struct Vapor	Struct App
GET /api/admins	Retorna todos os Admins	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Id: Int Name: String LastName: String dateOfBirth: String userID: Int
GET /api/admins/{id}	Retorna o Admin com o ID inserido	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Id: Int Name: String LastName: String dateOfBirth: String userID: Int
POST /api/admins	Adiciona um Admin. É necessário um ID de User válido. Crie primeiro um User com profile admins.	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Id: Int Name: String LastName: String dateOfBirth: String userID: Int
PATCH /api/admins/{id}	Atualiza o Admin do ID inserido	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Name: String LastName: String dateOfBirth: String userID: Int
DELETE /api/admins/{id}	Deleta o Admin do ID inserido. Retorna 200 OK caso tenha sido deletado com sucesso.		
GET /api/students	Retorna todos os Students	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Id: Int Name: String LastName: String dateOfBirth: String userID: Int
GET /api/students/{id}	Retorna o Student com o ID inserido	id: Int name: String lastName: String dateOfBirth: String classrooms: [Classroom] grades: [Grade] userID: User.ID	id: Int name: String lastName: String dateOfBirth: String classrooms: [Classroom] grades: [Grade] userID: Int
POST /api/students	Adiciona um Student. É necessário um ID de User válido. Crie primeiro um User com profile student.	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Id: Int Name: String LastName: String dateOfBirth: String userID: Int
PATCH /api/students/{id}	Atualiza o Student do ID inserido	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Name: String LastName: String dateOfBirth: String userID: Int
DELETE /api/students/{id}	Deleta o Student do ID inserido. Retorna 200 OK caso tenha sido deletado com sucesso.		

Endpoints	Descrição	Struct Vapor	Struct App
GET /api/instructors	Retorna todos os Instructors	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Id: Int Name: String LastName: String dateOfBirth: String userID: Int
GET /api/instructors/{id}	Retorna o Instructor do ID inserido.	id: Int name: String lastName: String dateOfBirth: String classrooms: [Classroom] userID: User.ID	id: Int name: String lastName: String dateOfBirth: String classrooms: [Classroom] userID: Int
POST /api/instructors	Adiciona um Instructor. É necessário um ID de User válido. Crie primeiro um User com profile instructor.	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Id: Int Name: String LastName: String dateOfBirth: String userID: Int
PATCH /api/instructors/{id}	Atualiza o Instructor do ID inserido.	Id: Int Name: String LastName: String dateOfBirth: String userID: User.ID	Name: String LastName: String dateOfBirth: String userID: Int
DELETE /api/instructors/{id}	Delete o Instructor do ID inserido. Retorna 200 OK caso tenha sido deletado com sucesso.		
GET /api/subjects	Retorna todos os Subjects	id: Int name: String	id: Int name: String
GET /api/subjects/{id}	Retorna o Subject do ID inserido.	id: Int name: String Classrooms: [Classrooms]	id: Int name: String Classrooms: [Classrooms]
POST /api/subjects	Adiciona um Subject.	id: Int name: String	id: Int name: String
PATCH /api/subjects/{id}	Atualizar o Subject do ID inserido.	id: Int name: String	name: String
DELETE /api/subjects/{id}	Delete o Subject do ID inserido. Retorna 200 OK caso tenha sido deletado com sucesso		
GET /api/grades	Retorna todos os Grades.	id: Int grades: [Double] finalGrade: Double studentID: Student.ID classroomID: Classroom.ID	id: Int grades: [Double] finalGrade: Double studentID: Int classroomID: Int

Endpoints	Descrição	Struct Vapor	Struct App
GET /api/grades/{id}	Retorna o Grade do ID inserido.	id: Int grades: [Double] finalGrade: Double studentID: Student.ID classroomID: Classroom.ID	id: Int grades: [Double] finalGrade: Double studentID: Int classroomID: Int
POST /api/grades	Adiciona um Grade. É necessário que o studentID e o classroomID sejam válidos.	id: Int grades: [Double] finalGrade: Double studentID: Student.ID classroomID: Classroom.ID	id: Int grades: [Double] finalGrade: Double studentID: Int classroomID: Int
PATCH /api/grades	Atualiza o Grade do ID inserido.	id: Int grades: [Double] finalGrade: Double studentID: Student.ID classroomID: Classroom.ID	grades: [Double] finalGrade: Double studentID: Int classroomID: Int
DELETE /api/grades	Deleta o Grade do ID inserido. Retorna 200 OK caso tenha sido deletado com sucesso.		
GET /api/classrooms	Retorna todos os Classrooms.	id: Int name: String subjectID: Subject.ID instructorID: Instructor.ID active: Bool	id: Int name: String subjectID: Int instructorID: Int active: Bool
GET /api/classrooms/{id}	Retorna o Classroom do ID inserido.	id: Int name: String subjectID: Subject.ID instructorID: Instructor.ID Grades: [Grade] instructor: Instructor subject: Subject students: [Student] active: Bool	id: Int name: String subjectID: Int instructorID: Int Grades: [Grade] instructor: Instructor subject: Subject students: [Student] active: Bool
POST /api/classrooms	Adiciona um Classroom. É necessário que o subjectID e o instructorID sejam válidos.	id: Int name: String subjectID: Subject.ID instructorID: Instructor.ID active: Bool	id: Int name: String subjectID: Int instructorID: Int active: Bool
PATCH /api/classrooms/{id}	Atualiza o Classroom do ID inserido.	id: Int name: String subjectID: Subject.ID instructorID: Instructor.ID active: Bool	name: String subjectID: Int instructorID: Int active: Bool
DELETE /api/classrooms/{id}	Deleta o Classroom do ID inserido. Retorna 200 OK caso tenha sido deletado com sucesso.		

Endpoints	Descrição	Struct Vapor	Struct App
GET /api/studentsClassrooms	Retorna todos os StudentsClassrooms.	id: Int studentID: Student.ID ClassroomID: Classroom.ID	id: Int studentID: Int ClassroomID: Int
GET /api/studentsClassrooms/{id}	Retorna o StudentClassrom do ID inserido.	id: Int studentID: Student.ID ClassroomID: Classroom.ID	id: Int studentID: Int ClassroomID: Int
POST /api/studentsClassrooms	Adiciona um StudentClassroom. É necessário que o studentID e o classroomID sejam válidos.	id: Int studentID: Student.ID ClassroomID: Classroom.ID	id: Int studentID: Int ClassroomID: Int
PATCH /api/studentsClassrooms/{id}	Atualiza o StudentClassroom do ID inserido	id: Int studentID: Student.ID ClassroomID: Classroom.ID	studentID: Int ClassroomID: Int
DELETE /api/studentsClassrooms/{id}	Delete o StudentClassrom do ID inserido.		
GET /api/users	Retorna todos os Users. Sem Password.	id: Int Username: String password: String profile: String	id: Int Username: String profile: String
GET /api/users/{id}	Retorna o User do ID inserido. Requer autenticação. Basic Authentication com username e password do User.	id: Int Username: String password: String profile: String	id: Int Username: String password: String profile: String
POST /api/users	Adiciona um User.	id: Int Username: String password: String profile: String	id: Int Username: String password: String profile: String
PATCH /api/users/{id}	Atualiza o User do ID inserido.	id: Int Username: String password: String profile: String	Username: String password: String profile: String
DELETE /api/users/{id}	Deleta o User do ID inserido. Retorna 200 OK caso tenha sido realizado com sucesso.		