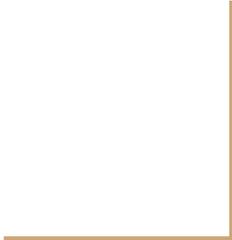




Projeto Ensalamento



CI1163
Departamento de Informática
Universidade Federal do Paraná
Curitiba – PR

2019

Projeto Ensalamento

- Versão 1 disponível
 - <http://ensalamento.c3sl.ufpr.br/>
- Projeto de arquitetura
 - Back-end e front-end
 - Ruby On Rails: <https://rubyonrails.org/>
 - Rápida prototipagem
 - Armazenamento
 - Banco de Dados PostgreSQL: <https://www.postgresql.org/>
 - Cálculo
 - Implementação de algoritmo em C

Exemplo da interface : baseada em CRUD

C3SL Consultar Universidade Mapa Entrar

Informações Públicas

Blocos

Cursos

Departamentos

Disciplinas

Professores

Salas

Salas/Biológicas - Uso Semanal

Salas/Humanas - Uso Semanal

Salas/Politécnico - Uso Semanal

Setores

Tipos de Salas

Turmas

DESIGN DE SOFTWARE - a;a;a;

Tamanho: 53

Professor(a): [Marcos Didonet Del Fabro](#)

Página: <http://web.inf.ufpr.br/didonet>

Alterações

Aulas

Data	Sala	Tema
------	------	------

Comentários

0 comentários Classificar por [Mais antigos](#)

 Adicione um comentário...

 [Plugin de comentários do Facebook](#)

NOVO Projeto Ensalador

- *Porque é necessário fazer um projeto novo ?*
 - Usado por outros setores/departamentos
 - Novas demandas solicitadas (MUITAS!)
 - Versão inicial com muitas modificações
 - Difícil **manutenção**

- Mas... implementar tudo do zero?
 - Modelo de classes/banco de dados (**sim**)
 - Interface (**sim**)
 - Código servidor (**sim**)
 - Cálculo do ensalamento (**não!**)

Decisões de projeto

- Acesso aos dados
 - Banco de dados relacional PostgreSQL (mesma escolha da versão 1)
 - OLTP (OnLine Analytical Processing)
 - Controle de transações, concorrência, segurança, persistência, ACID, etc. (disciplina de BD)
- Código servidor
 - Framework Loopback 3.0: <https://loopback.io/lb3>
 - Uma das funções: gerador de API com operações CRUD
 - Implementação de API RESTful em Node.js
 - Node.js: runtime Javascript (assíncrono) para a execução de código servidor
- Código cliente (interface)
 - React (<https://pt-br.reactjs.org/>): biblioteca Javascript para criação de interfaces

Gestão do desenvolvimento do projeto

- Ferramenta de gerenciamento de repositório *GitLab*
<https://gitlab.c3sl.ufpr.br/ensalamento/ensalamento/>
 - Código armazenado em servidor, com backup
 - Gestão de *commits*
 - log de alterações
 - Fusão de conflitos (*merge*)
 - Gestão de tarefas : *boards*
 - Continuous Integration (*CI*) / Continuous Delivery (*CD*)

Exemplo de interface

The screenshot shows the GitLab web interface for a project named 'ensalamento-back'. The top navigation bar includes the GitLab logo, 'Projects', 'Groups', 'Snippets', and 'Help'. A search bar and a user profile icon are also present. The left sidebar contains a navigation menu with options: Project, Details, Activity, Releases, Cycle Analytics, Repository, Issues (14), Merge Requests (1), CI / CD, Wiki, Snippets, and Members. The main content area displays the project details for 'ensalamento-back' (Project ID: 4178). It shows the license as GNU GPLv3, 56 Commits, 9 Branches, 0 Tags, and 2.3 MB Files. A yellow progress bar is visible. Below this, there is a dropdown menu for the current branch ('development') and a 'Merge branch '36-bug-path-does-not-exist-tmp' into 'development'' action, authored by Matheus Horstmann 5 months ago. A 'README' button is also present. At the bottom, a table lists the project's files and their last commit details.

Name	Last commit	Last update
bin	Fix #31: fix ensalador submodule	7 months ago
client	#4: Add .gitignore and initial project	1 year ago
common/models	Resolve "Preparar sistema de upload de relatórios"	6 months ago
server	Resolve "Bug: Path does not exist: tmp"	5 months ago
.dockerignore	Create Disciplina and EquivalenciaDisciplina Model	1 year ago
.env	Resolve "Preparar para conexão com Postgres"	1 year ago
.eslintignore	#4: Add .gitignore and initial project	1 year ago
.eslintrc	#4: Add .gitignore and initial project	1 year ago

Exemplo de tarefas

The image shows a Kanban board interface with a sidebar on the left and a main workspace. The sidebar contains navigation options: Project, Repository, Issues (14), Boards, Labels, Milestones, Merge Requests (1), CI / CD, Wiki, Snippets, and Members. The main workspace is divided into three columns: 'Open', 'To Do', and 'Doing'. The 'Open' column has 9 items, 'To Do' has 3, and 'Doing' has 3. Each task card includes a title, a number, and a status label.

Development Search or filter results...

Open 9 +

- Criar testes automatizados (atualização) #12
- Criar testes automatizados (deleção) #11
- Criar testes automatizados (leitura) #10
- Estudar modelos de testes automatizados #9
- Criar testes automatizados (inserção) #8
- Inserir constraint Unique Em EquivalenciaDisciplina #14

To Do 3

- Implementar subsistema de publicação do ensalamento #30
- Inserir validação de dados #19
- Atualizar o script de ensalamento #29

Doing 3

- Criar os Roles necessários #3
- Estudar API de documentação #20

Framework de desenvolvimento

- Editor : escolha livre (Atom, vim, nano, gedit, etc.)
- Geração da API e acesso ao banco: Loopback 3.0
 - Criação do modelo de classes através de comandos específicos
 - Geração do código base
 - Modelo de classes
 - API
 - Geração do acesso ao banco
 - Por herança ou novos métodos
 - Problema de incompatibilidade objeto-relacional (*object relational impedance mismatch*)

Criação do modelo através do framework

```

Create models
$ lb model

The generator guides you through
creating your model. Enter the values
highlighted in green. To accept the
default, just press Enter.

[?] Enter the model name: person
[?] Select the data-source to attach person to: db (memory)
[?] Select model's base class (PersistedModel)
[?] Expose person via the REST API? Yes
[?] Custom plural form (used to build REST URL): people
[?] Common model or server only? common
Let's add some person properties now.

Define a firstname property for the
person model

Enter an empty property name when done.
[?] Property name: firstname

Hit Enter to accept the default string
type:

[?] Property type: (Use arrow keys)
> string
  number
  boolean
  object
  array
  date
  buffer
  geopoint
  any
  (other)

```

Exemplo de JSON gerado

disciplina.json 2.93 KB

```
1 {
2   "name": "Disciplina",
3   "plural": "disciplinas",
4   "base": "PersistedModel",
5   "idInjection": false,
6   "properties": {
7     "codigo": {
8       "type": "String",
9       "id": true,
10      "required": true,
11      "length": 64,
12      "precision": null,
13      "scale": null
14    },
15    "nome": {
16      "type": "String",
17      "length": 256,
18      "precision": null,
19      "scale": null
20    },
21    "carga_horaria": {
22      "type": "Number",
23      "required": true,
24      "precision": null,
25      "scale": null,
26      "default": 0
27    },
28    "duracao": {
29      "type": "String",
30      "required": true,
31      "length": 64,
32      "precision": null,
33      "scale": null,
34      "default": "semestral"
35    },
36    "modalidade": {
37      "type": "String",
38      "length": 64,
39      "precision": null,
40      "scale": null,
41      "default": "presencial"
42    },

```

Exemplo de código modificado

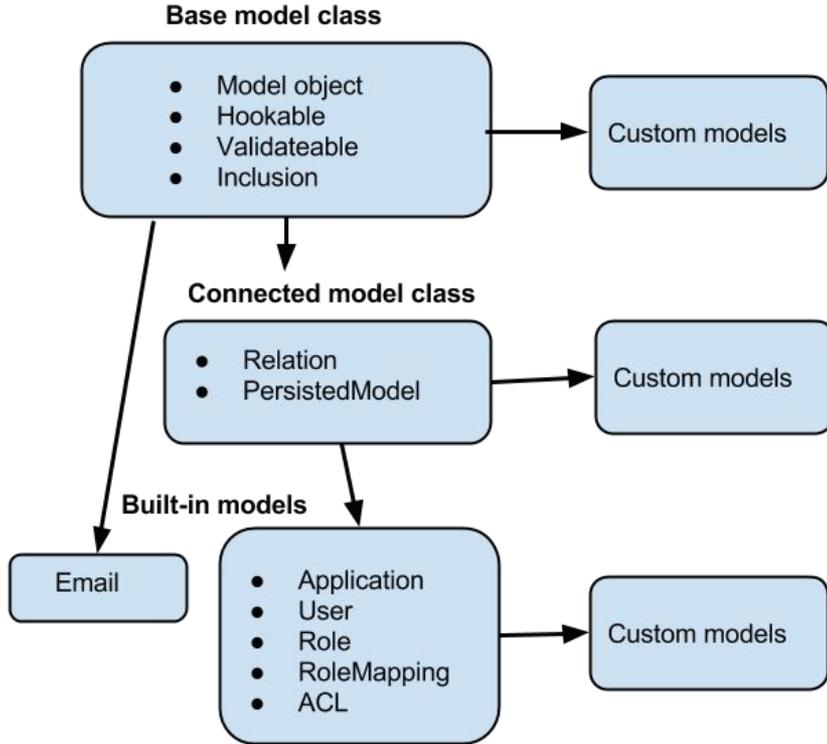
E ensalamento-back

- Project
- Repository
- Files
- Commits
- Branches
- Tags
- Contributors
- Graph
- Compare
- Charts
- Issues 14
- Merge Requests 1
- CI / CD
- Wiki
- Snippets
- Members

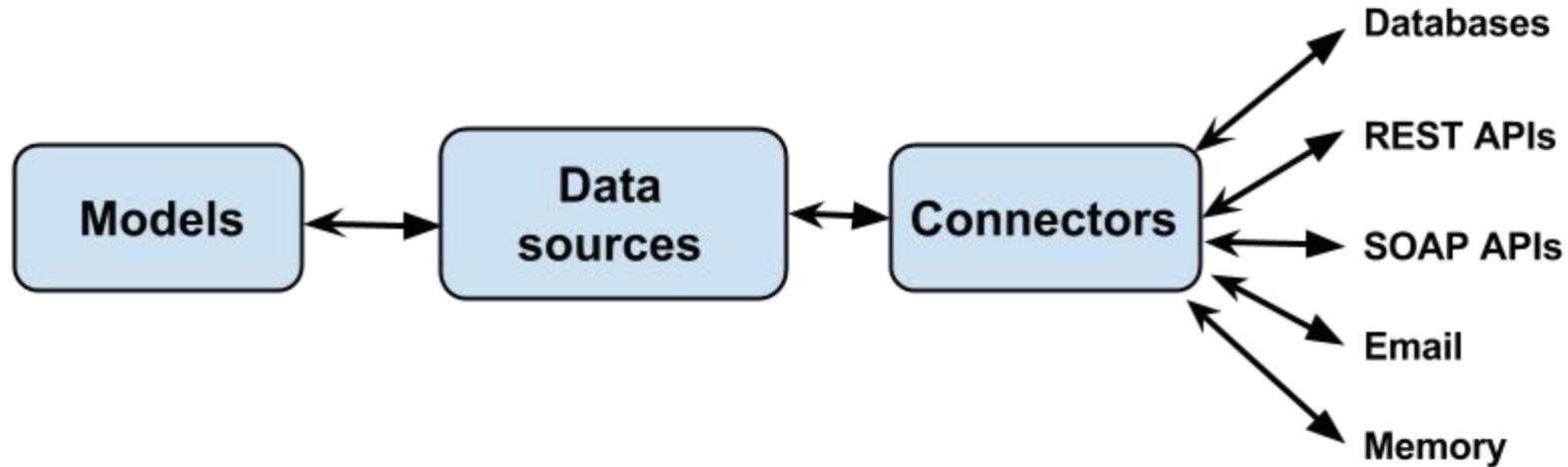
disciplina.js 1.97 KB

```
1 'use strict';
2
3 var app = require('../../server/server');
4
5 module.exports = function(Disciplina) {
6   // The follow remote methods was constructed because there is many problems
7   // in the Model Self Relation in LoopBack API 3
8   // The model EquivalenciaDisciplina was created to implement this relation
9
10  Disciplina.prototype.addEquivalencia = function(disciplina_eq_codigo, cb) {
11    var Eq = app.models.Equivalenciadisciplina;
12
13    Eq.create({"disciplinal": this.codigo, "disciplina2":disciplina_eq_codigo}, function(err, eq){
14      cb(err,eq);
15    });
16
17  };
18
19  Disciplina.prototype.getEquivalencias = function(cb) {
20    var Eq = app.models.Equivalenciadisciplina;
21    let myCode = this.codigo;
22    Eq.find({where:{or:[ {disciplinal:myCode}, {disciplina2:myCode} ]}}, function(err, eq){
23
24      // Filter only the ids of equivalent courses
25      // The id can be on disciplinal or disciplina2 columns
26      var ret = eq.map(function(x){
27
28        var codigoDisciplinal = x.__data.disciplinal;
29        var codigoDisciplina2 = x.__data.disciplina2;
30
31        if(myCode == codigoDisciplinal) return codigoDisciplina2;
32        return codigoDisciplinal;
33      });
34
35      // Remove repeated information
```

Vários tipos de modelos



Várias fontes de dados possíveis



API acessível

Bloco	Show/Hide	List Operations
container	Show/Hide	List Operations
Curso	Show/Hide	List Operations
Departamento	Show/Hide	List Operations
Disciplina	Show/Hide	List Operations
PATCH	/disciplinas	Patch an existing model instance or insert a new o
GET	/disciplinas	Find all instances of the model matched by filt
PUT	/disciplinas	Replace an existing model instance or insert a new o
POST	/disciplinas	Create a new instance of the model and persis
PATCH	/disciplinas/{id}	Patch attributes for a model instance and persis
GET	/disciplinas/{id}	Find a model instance by {id}
HEAD	/disciplinas/{id}	Check whether a model instance ex
PUT	/disciplinas/{id}	Replace attributes for a model instance and persis
DELETE	/disciplinas/{id}	Delete a model instance by {id}
GET	/disciplinas/{id}/departamento	Fetches belongsTo
GET	/disciplinas/{id}/equivalencias	Find all equivalenc
POST	/disciplinas/{id}/equivalencias	Create a new equivalence of the model and persis
DELETE	/disciplinas/{id}/equivalencias	Delete a equivalence of model instance by {id}
GET	/disciplinas/{id}/exists	Check whether a model instance ex
GET	/disciplinas/{id}/recursosnecessarios	Queries recursosne
POST	/disciplinas/{id}/recursosnecessarios	Creates a new instance in recursosne
DELETE	/disciplinas/{id}/recursosnecessarios	Deletes all recursosne
GET	/disciplinas/{id}/recursosnecessarios/{fk}	Find a related item by id fo
PUT	/disciplinas/{id}/recursosnecessarios/{fk}	Update a related item by id fo
DELETE	/disciplinas/{id}/recursosnecessarios/{fk}	Delete a related item by id fo

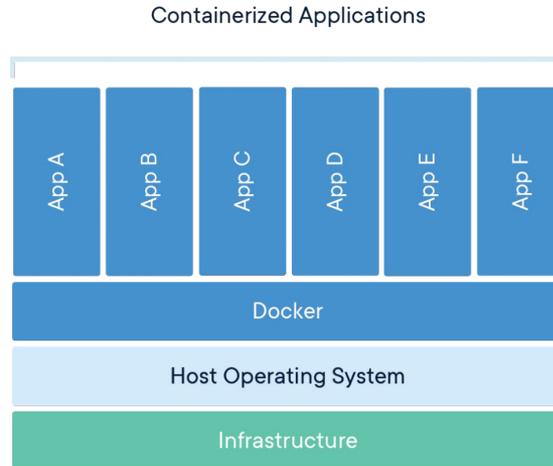
<http://ensalamentodev.c3sl.ufpr.br:3000/explorer/#/Disciplina>

E a implantação ? Isto é, será executado como?

- Criação de banco de produção
 - cópia inicial do banco de desenvolvimento
- Criação e instanciação do servidor da aplicação
 - Disponibilização da API
 - Disponibilização da interface gráfica
- Dois métodos comuns
 - “Na mão”: instanciar os diferentes servidores através de serviços ou scripts
 - Através de containers. Ex.: *docker*

Containers

- Criação, teste e implantação de aplicações em um ambiente separado.
 - Instalação de bibliotecas/versões/servidores próprios, mas na mesma máquina
 - O ambiente é isolado, mas usa as bibliotecas do sistema operacional



No ensalamento: *Docker*

- Container do servidor
- Container do banco de dados

- Estes serão “compostos”