
Desenvolvimento Ágil PPGI 2020 Documentation

Release 20.07

Carlo E. T. Oliveira

Aug 08, 2022

Contents:

1	Desenvolvimento Ágil PPGI 2020	1
2	Módulos Tutoriais	3
2.1	Jogo de Tabuleiro Quarto	3
2.2	Tutorial de Eventos	3
3	Modulos dos Participantes	5
3.1	Projeto Alpha	5
4	Laboratório de Automação de Sistemas Educacionais	7
5	Jogo de Tabuleiro Quarto.	9
5.1	Modulo Quarto	9
5.2	Classe Pino	9
5.3	Classe Casa	10
5.4	Classe Jogo do Quarto	11
5.5	Laboratório de Automação de Sistemas Educacionais	12
6	Projeto Alpha.	13
6.1	Modulo Main	13
6.2	Modulo Hotel	14
6.3	Modulo India	14
6.4	Laboratório de Automação de Sistemas Educacionais	15
7	Projeto Bravo.	17
7.1	Modulo Main	17
7.2	Modulo India	17
7.3	Modulo Hotel	17
7.4	Laboratório de Automação de Sistemas Educacionais	17
8	Projeto Charlie.	19
8.1	Modulo Main	19
8.2	Modulo Juliet	19
8.3	Modulo India	20
8.4	Laboratório de Automação de Sistemas Educacionais	20
9	Projeto Delta.	21
9.1	Modulo Main	21

9.2	Modulo Hotel	22
9.3	Modulo India	22
9.4	Modulo Juliet	22
9.5	Modulo Golf	22
9.6	Laboratório de Automação de Sistemas Educacionais	23
10	Projeto Echo.	25
10.1	Modulo Main	25
10.2	Modulo Hotel	25
10.3	Modulo India	25
10.4	Modulo Juliet	26
10.5	Modulo Golf	26
10.6	Laboratório de Automação de Sistemas Educacionais	26
11	Projeto Foxtrot.	27
11.1	Modulo Main	27
11.2	Modulo Hotel	27
11.3	Modulo India	27
11.4	Modulo Juliet	28
11.5	Modulo Golf	28
11.6	Laboratório de Automação de Sistemas Educacionais	28
12	Tutorial Um.	31
12.1	Modulo Tutorial Um	31
12.2	Classe Eventos	31
12.3	Laboratório de Automação de Sistemas Educacionais	32
13	Changelog	33
13.1	Unreleased	33
13.2	20.07	33
13.3	20.06	34
13.4	18.7	34
13.5	Laboratório de Automação de Sistemas Educacionais	34
14	Indices e tabelas	35
15	Laboratório de Automação de Sistemas Educacionais	37
	Python Module Index	39
	Index	41

Desenvolvimento Ágil PPGI 2020

Projetos da disciplina de Desenvolvimento Ágil Orientado a Objetos, utilizando a plataforma de programação SuperPython.

Desenvolveremos aplicações reais WEB nos seguintes temas: Tutorial Kwarwp: Um tutorial para aprendizado da linguagem Python através das aventuras de um índio. ArtAntCovid: Um aplicativo de acompanhamento e proteção da população contra a pandemia viral. ActivUFRJ Leve: Um ambiente virtual dedicado ao ensino e pesquisa com uma interface intuitiva. Plataforma SupyGirls: Uma plataforma para desenvolvimento de jogos educacionais em Python. Diversas ferramentas e bibliotecas de desenvolvimento serão usadas como o Pycharm, Jupyter, GitLab, Docker, Notion e outras. A ideia é ir do mais básico ao mais avançado em doze semanas.

Módulos desenvolvidos para ensinar algum aspecto da programação

2.1 Jogo de Tabuleiro Quarto

- Jogo do Quarto : As classes Pino, Casa e Tabuleiro

Jogo de Tabuleiro Quarto.

2.2 Tutorial de Eventos

- Tutorial : A classe Evento.

Tutorial Um.

Modulos dos Participantes

Desenvolvimento das equipes de programação

Projetos das Equipes:

3.1 Projeto Alpha

- Um pequeno labirinto com várias câmaras.

Projeto Alpha.

Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

Jogo de Tabuleiro Quarto.

5.1 Modulo Quarto

Jogo de Tabuleiro Quarto.

Code author: Carlo Oliveira <carlo@ufrj.br>

Classes neste módulo:

JogoDoQuarto Jogo de tabuleiro para dois jogadores.

Pino Representa um pino que se move de um tabuleiro para outro.

Casa Representa um case que pode receber um pino no tabueiro.

5.1.1 Changelog

New in version 20.07: Criando o tabuleiro.

New in version 20.07.01: Criando Pino e Casa.

See also:

Page *Desenvolvimento Ágil PPGI 2020*

Note: Este módulo é um tutorial ensinando a programar um jogo de tabuleiro simples.

5.2 Classe Pino

Módulo *Jogo de Tabuleiro Quarto*.

class `adda.quarto.Pino` (*indice, cor, casa, dx=101, dy=60*)

Pinos do jogo quarto variando em cor, altura, forma e buraco

Observação: Uma *folha sprite* é uma imagem contendo vários desenhos com peças que serão usados em um jogo.

Parameters

- **indice** – O índice do pino na *folha sprite* (cada imagem tem 8 pinos).
- **cor** – Nome da imagem sprite com um conjunto de pinos claros ou escuros.
- **casa** – Casa original onde o pino vai ser colocado.
- **dx** – o intervalo horizontal em pixels entre duas figuras de pino na *folha sprite*.
- **dy** – o intervalo vertical em pixels entre duas figuras de pino na *folha sprite*.

static pino (*nome*)

Retorna o pino localizado dado o nome.

Parameters **nome** – O nome do pino no dicionário de pinos *PINOS*.

Returns O objeto pino que tem o nome dado.

Return type Instância da classe Pino.

static pinos (*cor, table*)

Cria um conjunto de pinos de uma cor e aloca no dado tabuleiro.

Parameters

- **table** – A tabela contendo um conjunto de casas onde os pinos serão alocados.
- **cor** – Nome da imagem sprite com um conjunto de pinos claros ou escuros.

Returns Uma lista de pinos alocados nas casas do tabuleiro dado.

Return type Lista contendo instâncias da classe Pino.

5.3 Classe Casa

Módulo *Jogo de Tabuleiro Quarto*.

class `adda.quarto.Casa` (*base, fundo, x, y, dx=0, dy=3, mx=0, my=0, livre=False, t=75*)

Casa do jogo quarto, podendo ser do tabuleiro central ou adjacentes.

Parameters

- **base** – A cena em que este tabuleiro vai entrar.
- **fundo** – A imagem que representa esta casa.
- **livre** – Indica se a casa está livre ou ocupada quando for criada.
- **t** – Tamanho (altura e largura do desenho da casa, inicia em um oitavo do tabuleiro).
- **x** – A posição absoluta horizontal em casas que esta casa vai ser colocada.
- **y** – A posição absoluta vertical em casas que esta casa vai ser colocada.
- **dx** – O deslocamento horizontal em casas que esta casa tem em relação à esquerda.
- **dy** – O deslocamento vertical em casas que esta casa tem em relação ao topo.
- **mx** – Ajuste fino horizontal em pixels do posicionamento da casa.

- **my** – Ajuste fino vertical em pixels do posicionamento da casa.

Note:

folha sprite é uma imagem contendo vários desenhos com peças que serão usados em um jogo.

dropper dicionário necessário para definir o comportamento do drop. Cada chave deste dicionário deve ser o nome de um pino passível de ser arrastado para esta casa. Os nomes são do tipo *topo0* ... *topo7* ou *lado0* ... *lado7*. A construção `{f"{nome}{pino}": ..}` é chamada em Python de dict comprehension, pois gera um dicionário a partir de uma iteração. O termo `f"{nome}{pino}"` constrói as chaves do dicionário a partir da lista (topo, lado) e dos números de 0 a 7. O termo `lambda ev, nome_pino, _: self.entrar(nome_pino)` é uma função anônima que recebe os parâmetros *ev* e *nome_pino* enviados pelo mecanismo interno de arrasto do Elemento. Neste caso, para qualquer pino eu quero que ele entre na casa corrente, por isso o `self.entrar(nome_pino)` é chamado dentro da função anônima.

entrar (*nome*)

Permite um pino entrar nesta casa se ela estiver livre.

Parameters **nome** – O nome do pino que vai entrar na casa.

5.4 Classe Jogo do Quarto

Módulo *Jogo de Tabuleiro Quarto*.

class `adda.quarto.JogoDoQuarto`

Bases: `object`

Jogo do Quarto - Jogo de Tabuleiro. Quarto é um jogo de tabuleiro para dois jogadores inventado pelo matemático suíço Blaise Müller em 1991.

O jogo é jogado em um tabuleiro 4 × 4. Existem 16 peças únicas para brincar, cada uma das quais é:

- alto ou baixo;
- vermelho ou azul (ou um par diferente de cores, por exemplo, madeira manchada de clara ou escura);
- quadrado ou circular; e
- topo oco ou topo sólido.

Os jogadores se revezam na escolha de uma peça que o outro jogador deve colocar no tabuleiro. Um jogador vence colocando uma peça no tabuleiro que forma uma linha horizontal, vertical ou diagonal de quatro peças, todas com um atributo em comum (todas curtas, todas circulares, etc.). Uma regra de variante incluída em muitas edições fornece uma segunda maneira de ganhar colocando quatro peças correspondentes em um quadrado 2 × 2.

table (*base, fundo, linhas=5, colunas=5, dx=0, dy=3, mx=0, my=0, livre=False*)

Cria um conjunto de casas de um tipo e aloca na cena dada.

Parameters

- **base** – A cena em que este tabuleiro vai entrar.
- **fundo** – A imagem que representa cada casa.
- **livre** – Indica se as casas estão livres ou ocupadas quando forem criadas.
- **t** – Tamanho (altura e largura do desenho da casa, inicia em um oitavo do tabuleiro).

- **linhas** – O número de linhas de casas neste tabuleiro.
- **colunas** – O número de colunas de casas neste tabuleiro.
- **dx** – O deslocamento horizontal em casas que cada casa tem em relação à esquerda.
- **dy** – O deslocamento vertical em casas que cada casa tem em relação ao topo.
- **mx** – Ajuste fino horizontal em pixels do posicionamento de cada casa.
- **my** – Ajuste fino vertical em pixels do posicionamento de cada casa.

Returns Uma tabela de duas dimensões de casas arranjadas nas posições dadas.

Return type Lista de listas contendo instâncias da classe Casa.

vai ()

Aloca esta cena na página do navegador.

5.5 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

Um pequeno labirinto com várias câmaras.

6.1 Modulo Main

Projeto ALPHA - Jogo de Labirinto com cenas.

Calabouço de Barro.

Code author: Rodrigo Esquinelato <resquinelato@gmail.com>

Code author: Charles Soares Pimentel <pimentelufrij@gmail.com>

6.1.1 Changelog

Código alterado de Monica Novellino <monicanovellino@gmail.com>

New in version 20.07: Adicionei 5 imagens iniciais do labirinto e alterei o pacman (podem criar outro peronagem)
Contador adicionado para gerar as seguintes fases em função da linha da matriz Contador adicionado para gerar as seguintes posições iniciais em função do índice da lista

```
class alpha.main.Eventos
    Associa um evento a uma imagem e captura eventos de teclado.

    BONECO = 'https://i.imgur.com/k63kwfa.png'
    CENA_corredor_1 = 'https://i.imgur.com/L71ZV6Z.png'
    CENA_corredor_2 = 'https://i.imgur.com/5Qno2fs.png'
    CENA_corredor_3 = 'https://i.imgur.com/gZ5wc0h.png'
    CENA_corredor_4 = 'https://i.imgur.com/xI8i7Nc.png'
    CENA_corredor_5 = 'https://i.imgur.com/GLVctqb.png'
    anda_boneco (ev=None)
```

```
listaFase = ['https://i.imgur.com/5Qno2fs.png', 'https://i.imgur.com/gZ5wc0h.png', 'ht
vai ()
    mostra corredor do labirinto
```

See also:

Page *Desenvolvimento Ágil PPGI 2020*

6.2 Modulo Hotel

Início *Projeto Alpha*.

Projeto ALPHA - Jogo de Labirinto com cenas.

Code author: Monica Novellino <monicanovellino@gmail.com>

6.2.1 Changelog

New in version 0.1: - Planta da casa - Boneco andando com as setas

```
class alpha.hotel.Eventos
    Associa um evento a uma imagem e captura eventos de teclado.

    BONECO = 'https://upload.wikimedia.org/wikipedia/commons/thumb/6/62/Prick%C3%A4tarpu
    CENA_COZINHA = 'https://www.decorfacil.com/wp-content/uploads/2018/03/20180311tons-de-
    CENA_ESCRITORIO = 'https://www.decorfacil.com/wp-content/uploads/2018/03/20180311tons-
    CENA_PLANTA = 'https://w7.pngwing.com/pngs/38/713/png-transparent-floor-plan-design-pl
    anda_boneco (ev=None)
        ” Faz o boneco caminhar com a cptura das setas. :param ev: estrutura enviad pelo evento onde se recupera
        informações.

    vai ()
        mostra a cena da planta da casa.
```

6.3 Modulo India

Início *Projeto Alpha*.

Jogo de Labirinto com cenas.

Calabouço de Barro.

Code author: Rodrigo Esquinelato <resquinelato@gmail.com>

Classe neste módulo:

Eventos Gera a cena, personagem e a movimentação

6.3.1 Changelog

New in version 20.07: Criando a lógica do jogo e o layout das fases.

Contador adicionado para gerar as seguintes fases em função da linha da matriz.

Contador adicionado para gerar as seguintes posições iniciais em função do índice da lista.

```
class alpha.india.Eventos
```

```
    Associa um evento a uma imagem
```

```
    BONECO = 'https://i.imgur.com/k63kwfa.png'
```

```
    CENA_corredor_1 = 'https://i.imgur.com/L71ZV6Z.png'
```

```
    CENA_corredor_2 = 'https://i.imgur.com/5Qno2fs.png'
```

```
    CENA_corredor_3 = 'https://i.imgur.com/gZ5wc0h.png'
```

```
    CENA_corredor_4 = 'https://i.imgur.com/xI8i7Nc.png'
```

```
    CENA_corredor_5 = 'https://i.imgur.com/GLVctqb.png'
```

```
    anda_boneco (ev=None)
```

```
    listaFase = ['https://i.imgur.com/5Qno2fs.png', 'https://i.imgur.com/gZ5wc0h.png', 'ht
```

```
    vai ()
```

```
    Returns Cenário do labirinto.
```

6.4 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

Um jogo que ensine o básico da programação no SuperPython.

7.1 Modulo Main

See also:

Page *Desenvolvimento Ágil PPGI 2020*

7.2 Modulo India

Início *Projeto Bravo*.

7.3 Modulo Hotel

Início *Projeto Bravo*.

Em Construção.

7.4 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação de Sistemas Educacionais

Quebra-cabeças para ensino de programação orientada a objeto.

8.1 Modulo Main

Charlie - Projeto sem descrição, (mude esta linha).

Code author: Isaac D' Césares (golf) <isaac.dcesares@gmail.com>

Code author: Nome Sobrenome (hotel) <mail@local.tipo>

Code author: Nome Sobrenome (india) <mail@local.tipo>

Code author: Nome Sobrenome (juliet) <mail@local.tipo>

8.1.1 Changelog

New in version 20.07: Descreva o que você adicionou no código.

See also:

Page *Desenvolvimento Ágil PPGI 2020*

8.2 Modulo Juliet

Início *Projeto Charlie*.

ensinar orientando ao objeto com quebra cabeça .

Code author: Leniah Lima <leniah@nce.ufrj.br>

8.2.1 Changelog

New in version 20.07: Descreva o que você adicionou no código.

class charlie.juliet.minipython (*esta_cena, chama_quando_acerta*)

Usa um editor de imagem (/) e recorta o Herdograma em linhas geracionais. No game, o jogador terá que clicar nas linhas em ordem certa para montar o herdograma corretamente.

posiciona_proxima (*posicao*)

Chamdo pelo clique (vai) de cada peça. Atualiza a próxima posição da peça. Calcula se montou correto, comparando com a lista de posições corretas. Se já montou quatro peças, e não acerto sinaliza com zero, para iniciar o jogo.

class charlie.juliet.normal

mostra_conteudo_minipython (*_)

vai (*_)

8.3 Modulo India

Início *Projeto Charlie*.

Em Construção.

8.4 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

JOGO DA MEMÓRIA.

9.1 Modulo Main

Delta - Projeto sem descrição, (mude esta linha).

Code author: Paulo Assumpção (golf) <mail@local.tipo>

Code author: Anni Provietti (hotel) <mail@local.tipo>

Code author: Emanuelle Simas (india) <ellesimas@gmail.com>

Code author: Nome Sobrenome (juliet) <mail@local.tipo>

9.1.1 Changelog

New in version 20.07: Descreva o que você adicionou no código.

```
class delta.main.end
```

```
class delta.main.start
```

 Gera uma classe onde há a tela de partida do jogo da memória.

```
    chama (event=None)
```

```
    solucao = None
```

 O bind liga um evento a um método

```
    vai ()
```

See also:

Page *Desenvolvimento Ágil PPGI 2020*

9.2 Modulo Hotel

Início *Projeto Delta*.

```
class delta.hotel.Card (name, image, position, cena, rule)
```

```
    turnDown (env=None)
```

```
    turnUp (env=None)
```

```
class delta.hotel.Game
```

```
    cena = <Cena id='140719324405968'>
```

```
    create_2x2_cards ()
```

```
        matrix 2x3: 1A 1B 2A 2B
```

```
    previous_selected_card = None
```

```
    static rule (selected_card)
```

```
    shuffle_cards ()
```

```
    vai ()
```

9.3 Modulo India

Início *Projeto Delta*.

9.4 Modulo Juliet

Início *Projeto Delta*.

9.5 Modulo Golf

Início *Projeto Delta*.

Jogo da Memória.

Code author: Paulo Assumpção <paulo.assump@gmail.com>

9.5.1 Changelog

New in version 20.07: Grid 2x5 de cartões do jogo da memória shuffle das cartas bind do click sobre o botão regra do jogo

```
class delta.golf.Card (name, image, position, cena, rule)
```

```
    turnDown (env=None)
```

```
    turnUp (env=None)
```

```
class delta.golf.Game

    cena = <Cena id='140719323680016'>
    create_2x5_cards ()
    previous_selected_card = None
    static rule (selected_card)
    shuffle_cards ()
    vai ()

delta.golf.IMG_HEIGHT = 150
    Classe responsável pelo desenho das cartas na cena
```

9.6 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

jogo de tabuleiro de Olimpíada Matemática.

10.1 Modulo Main

Jogo Olimpíadas de Matemática20.07 * NEW: O jogo original Autor: Grupo Echo (Aline, Lorena, Renato e Victória)

```
class echo.main.Jogo
    Representa uma cena de tabuleiro com o botão
```

```
    vai ()
        Mostra o tabuleiro
```

```
class echo.main.Tabuleiro
    Representa uma cena da calçada da praia
```

```
    vai ()
        Mostra a cena do tabuleiro
```

See also:

Page *Desenvolvimento Ágil PPGI 2020*

10.2 Modulo Hotel

Início *Projeto Echo*.

10.3 Modulo India

Início *Projeto Echo*.

Projeto sem descrição, (mude esta linha).

Code author: Victória Regina Caruzo <victoriareginalattes@gmail.com>

10.3.1 Changelog

New in version 20.07: Descreva o que você adicionou no código.

10.4 Modulo Juliet

Início *Projeto Echo*.

10.5 Modulo Golf

Início *Projeto Echo*.

10.6 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

Um jogo que ensine o básico da programação no SuperPython - SuperPython World.

11.1 Modulo Main

Foxtrot - Projeto sem descrição, (mude esta linha).

Code author: Nome Sobrenome (golf) <mail@local.tipo>

Code author: Nome Sobrenome (hotel) <mail@local.tipo>

Code author: Nome Sobrenome (india) <mail@local.tipo>

Code author: Nome Sobrenome (juliet) <mail@local.tipo>

11.1.1 Changelog

New in version 20.07: Descreva o que você adicionou no código.

See also:

Page *Desenvolvimento Ágil PPGI 2020*

11.2 Modulo Hotel

Início *Projeto Foxtrot.*

11.3 Modulo India

Início *Projeto Foxtrot.*

11.4 Modulo Juliet

Início *Projeto Foxtrot*.

11.5 Modulo Golf

Início *Projeto Foxtrot*.

Projeto SuperPython World,

Code author: Raquel Fernandes <raquelmachado4993@gmail.com>

11.5.1 Changelog

New in version 20.07: Página inicial do Jogo.

```
foxtrot.golf.FOCO = 'https://i.imgur.com/6e096Va.png'  
    criando classe jogo
```

```
class foxtrot.golf.Jogo
```

```
    avatar2 = None  
        exibindo elementos na cena sobre
```

```
    botao_sobre = None  
        exibindo cena
```

```
    capa = None  
        inserindo elementos na cena
```

```
    castelo = None  
        texto castelo
```

```
    cena_sobre = None  
        Cena sobre
```

```
    cenaauthor = None  
        ação após escolha do Avatar
```

```
    porta = None  
        textos
```

```
    texto_jogar = None  
        ação caso seja clicado
```

11.6 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação de Sistemas Educacionais

12.1 Modulo Tutorial Um

Tutorial Um - respondendo dúvidas do SuperPython.

Code author: Carlo Oliveira <carlo@ufrj.br>

- Como associar um evento a uma imagem
- Como combinar cenas em salas diferentes
- Como capturar o teclado

Classes neste modulo:

Eventos Exemplo de capturar eventos de mouse e teclado.

12.1.1 Changelog

New in version 20.07: Documentação do tutorial.

New in version 20.07.1: Inclui um import.

See also:

Page *Desenvolvimento Ágil PPGI 2020*

Note: Este módulo é um tutorial ensinando a programar um jogo de tabuleiro simples.

12.2 Classe Eventos

Módulo *Tutorial Um*.

class adda.tutorial_um.Eventos

Bases: object

Associa um evento a uma imagem e captura eventos de teclado.

anda_banhista (*ev=None*)

” Faz o banhista caminhar com a captura das setas.

Parameters **ev** – estrutura enviada pelo evento onde se recupera informações.

vai ()

mostra a cena da calçada.

ve_dark (*ev=None*)

” Faz o letreiro mostrar ou ocultar quando se passa o mouse no banhista.

Parameters **ev** – estrutura enviada pelo evento onde se recupera informações.

12.3 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#).

13.1 Unreleased

- to be defined

13.2 20.07

13.2.1 Fixed

- Shows by default main.py at page title.
- Some PEP8 refactoring and minor documentantion.

13.2.2 Added

- Handles filenames other than main.py.
- Accept new file specification in URL, use *<filename>.pys*
- New CHANGELOG.rst
- New readme.rst
- Add Version to inner pages

13.3 20.06

13.3.1 Fixed

- Kwarwp assignment documentation.
- Docker file gunicorn configuration

13.3.2 Added

- New readme.md

13.4 18.7

13.4.1 Added

- Original Code.
-

13.5 Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

CHAPTER 14

Indices e tabelas

- genindex
- modindex
- search

CHAPTER 15

Laboratório de Automação de Sistemas Educacionais

Copyright © Carlo Olivera

LABASE - NCE - UFRJ



Laboratório de Automação
de Sistemas Educacionais

a

adda.quarto (*Web*), 9
adda.tutorial_um (*Web*), 31
alpha.hotel (*Web*), 14
alpha.india (*Web*), 14
alpha.main (*Web*), 13

c

charlie.juliet (*Web*), 19
charlie.main (*Web*), 19

d

delta.golf (*Web*), 22
delta.hotel (*Web*), 22
delta.main (*Web*), 21

e

echo.india (*Web*), 25
echo.main (*Web*), 25

f

foxtrot.golf (*Web*), 28
foxtrot.main (*Web*), 27

A

adda.quarto (*module*), 9
 adda.tutorial_um (*module*), 31
 alpha.hotel (*module*), 14
 alpha.india (*module*), 14
 alpha.main (*module*), 13
 anda_banhista () (*adda.tutorial_um.Eventos method*), 32
 anda_boneco () (*alpha.hotel.Eventos method*), 14
 anda_boneco () (*alpha.india.Eventos method*), 15
 anda_boneco () (*alpha.main.Eventos method*), 13
 avatar2 (*foxtrot.golf.Jogo attribute*), 28

B

BONECO (*alpha.hotel.Eventos attribute*), 14
 BONECO (*alpha.india.Eventos attribute*), 15
 BONECO (*alpha.main.Eventos attribute*), 13
 botao_sobre (*foxtrot.golf.Jogo attribute*), 28

C

capa (*foxtrot.golf.Jogo attribute*), 28
 Card (*class in delta.golf*), 22
 Card (*class in delta.hotel*), 22
 Casa (*class in adda.quarto*), 10
 castelo (*foxtrot.golf.Jogo attribute*), 28
 cena (*delta.golf.Game attribute*), 23
 cena (*delta.hotel.Game attribute*), 22
 CENA_corredor_1 (*alpha.india.Eventos attribute*), 15
 CENA_corredor_1 (*alpha.main.Eventos attribute*), 13
 CENA_corredor_2 (*alpha.india.Eventos attribute*), 15
 CENA_corredor_2 (*alpha.main.Eventos attribute*), 13
 CENA_corredor_3 (*alpha.india.Eventos attribute*), 15
 CENA_corredor_3 (*alpha.main.Eventos attribute*), 13
 CENA_corredor_4 (*alpha.india.Eventos attribute*), 15
 CENA_corredor_4 (*alpha.main.Eventos attribute*), 13
 CENA_corredor_5 (*alpha.india.Eventos attribute*), 15
 CENA_corredor_5 (*alpha.main.Eventos attribute*), 13
 CENA_COZINHA (*alpha.hotel.Eventos attribute*), 14
 CENA_ESCRITORIO (*alpha.hotel.Eventos attribute*), 14

CENA_PLANTA (*alpha.hotel.Eventos attribute*), 14
 cena_sobre (*foxtrot.golf.Jogo attribute*), 28
 cenaauthor (*foxtrot.golf.Jogo attribute*), 28
 chama () (*delta.main.start method*), 21
 charlie.juliet (*module*), 19
 charlie.main (*module*), 19
 create_2x2_cards () (*delta.hotel.Game method*), 22
 create_2x5_cards () (*delta.golf.Game method*), 23

D

delta.golf (*module*), 22
 delta.hotel (*module*), 22
 delta.main (*module*), 21

E

echo.india (*module*), 25
 echo.main (*module*), 25
 end (*class in delta.main*), 21
 entrar () (*adda.quarto.Casa method*), 11
 Eventos (*class in adda.tutorial_um*), 31
 Eventos (*class in alpha.hotel*), 14
 Eventos (*class in alpha.india*), 15
 Eventos (*class in alpha.main*), 13

F

FOCO (*in module foxtrot.golf*), 28
 foxtrot.golf (*module*), 28
 foxtrot.main (*module*), 27

G

Game (*class in delta.golf*), 22
 Game (*class in delta.hotel*), 22

I

IMG_HEIGHT (*in module delta.golf*), 23

J

Jogo (*class in echo.main*), 25

Jogo (*class in* *foxtrot.golf*), 28
JogoDoQuarto (*class in* *adda.quarto*), 11

L

listaFase (*alpha.india.Eventos attribute*), 15
listaFase (*alpha.main.Eventos attribute*), 14

M

minipython (*class in* *charlie.juliet*), 20
mostra_conteudo_minipython() (*charlie.juliet.normal method*), 20

N

normal (*class in* *charlie.juliet*), 20

P

Pino (*class in* *adda.quarto*), 9
pino() (*adda.quarto.Pino static method*), 10
pinos() (*adda.quarto.Pino static method*), 10
porta (*foxtrot.golf.Jogo attribute*), 28
posiciona_proxima() (*charlie.juliet.minipython method*), 20
previous_selected_card (*delta.golf.Game attribute*), 23
previous_selected_card (*delta.hotel.Game attribute*), 22

R

rule() (*delta.golf.Game static method*), 23
rule() (*delta.hotel.Game static method*), 22

S

shuffle_cards() (*delta.golf.Game method*), 23
shuffle_cards() (*delta.hotel.Game method*), 22
solucao (*delta.main.start attribute*), 21
start (*class in* *delta.main*), 21

T

table() (*adda.quarto.JogoDoQuarto method*), 11
Tabuleiro (*class in* *echo.main*), 25
texto_jogar (*foxtrot.golf.Jogo attribute*), 28
turnDown() (*delta.golf.Card method*), 22
turnDown() (*delta.hotel.Card method*), 22
turnUp() (*delta.golf.Card method*), 22
turnUp() (*delta.hotel.Card method*), 22

V

vai() (*adda.quarto.JogoDoQuarto method*), 12
vai() (*adda.tutorial_um.Eventos method*), 32
vai() (*alpha.hotel.Eventos method*), 14
vai() (*alpha.india.Eventos method*), 15
vai() (*alpha.main.Eventos method*), 14
vai() (*charlie.juliet.normal method*), 20

vai() (*delta.golf.Game method*), 23
vai() (*delta.hotel.Game method*), 22
vai() (*delta.main.start method*), 21
vai() (*echo.main.Jogo method*), 25
vai() (*echo.main.Tabuleiro method*), 25
ve_dark() (*adda.tutorial_um.Eventos method*), 32