

# *Sistemas Reativos*

**reactive:**

*readily responsive to a stimulus*

(Merriam-Webster)

*Francisco Sant'Anna*  
`francisco@ime.uerj.br`



# Classificação de “Sistemas”

(Berry 1989)

- Sistemas transformacionais
  - *“compute results from a given set of inputs”*
- Sistemas interativos
  - *“interact at their own speed with users or with other programs”*
- Sistemas reativos
  - *“interact with their environment, but at a speed which is determined by the environment, not by the program itself”*
- gcc, gdb, quake, md5, shell, zip, http, gui

# Environment / Ambiente

- Componente externo à aplicação
  - “mundo analógico” no qual a aplicação está inserida
- Input / Sensores
  - Teclado, Mouse, Botão, Sensores (luminosidade, temperatura, etc.), Temporizador
- Output / Atuadores
  - LED, Tela, Impressora, Motor
- Input  $\Leftrightarrow$  Output
  - Serial, Rádio

# Videos

# Projetos

- Arduino / C
  - Sistemas embarcados
- SDL / C
  - Jogos, aplicações gráficas
- Céu (Arduino, SDL)

```

ArmageddonButton::ArmageddonButton(<...>):
    RectComponent(<...>),
    <...>
    pressed(false); // initial button state
    press_time();    // how long since the 1st click
    <...>
{
    <...>
}

void ArmageddonButton::draw (<...>) {
    <...>
}

void ArmageddonButton::update (float delta) {
    <...>
    if (pressed) {
        press_time += delta;
        if (press_time > 1.0f) {
            pressed = false;    // giving up, 1st
            press_time = 0;      // too
        }
    } else {
        <...>
        press_time = 0;
    }
}

void ArmageddonButton::on_click (<...>) {
    if (pressed) {
        server->send_armageddon_event();
    } else {
        pressed = true;
    }
}

```



```

class ArmageddonButton with
    <...>
do
    var RectComponent component = <...>;
    <...>
    loop do
        await component.on_click;
        watching is do
            await component.on_click;
            break;
        end
    end
    <...>
    emit global:go_armageddon;
end

```