

AGENDEJÁ

SISTEMA DISTRIBUÍDO HETEROGÊNEO

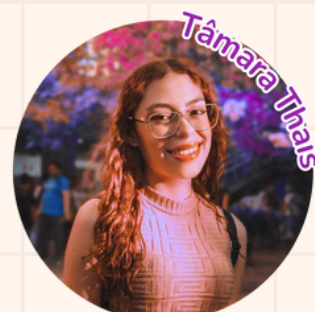
Integração REST (Django), SOAP (Java) e WebSocket via API Gateway.



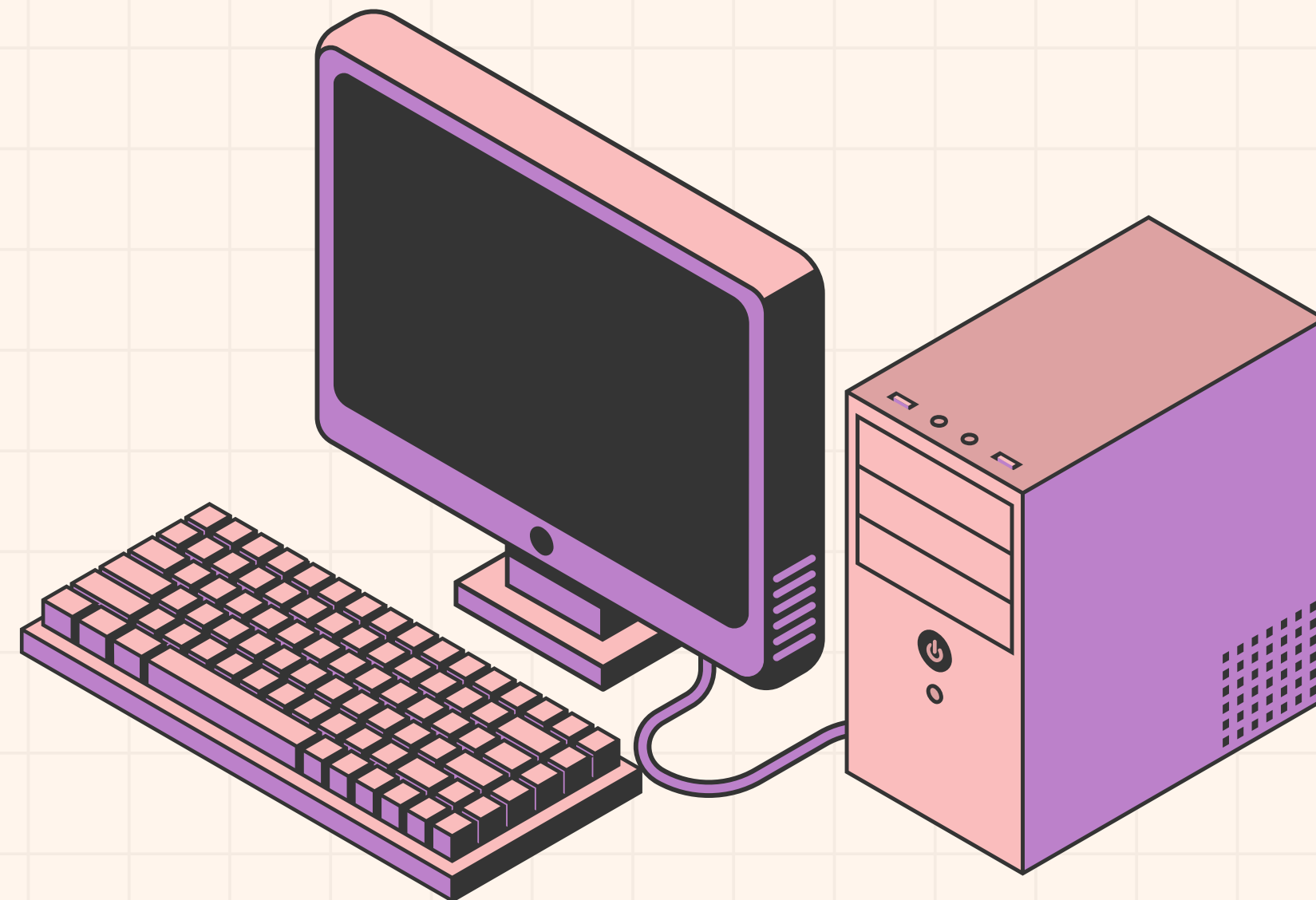
Emanuelly Karine



Rebeca Noemi



Tâmara Thais



VISÃO GERAL DO SISTEMA

O que desenvolvemos:

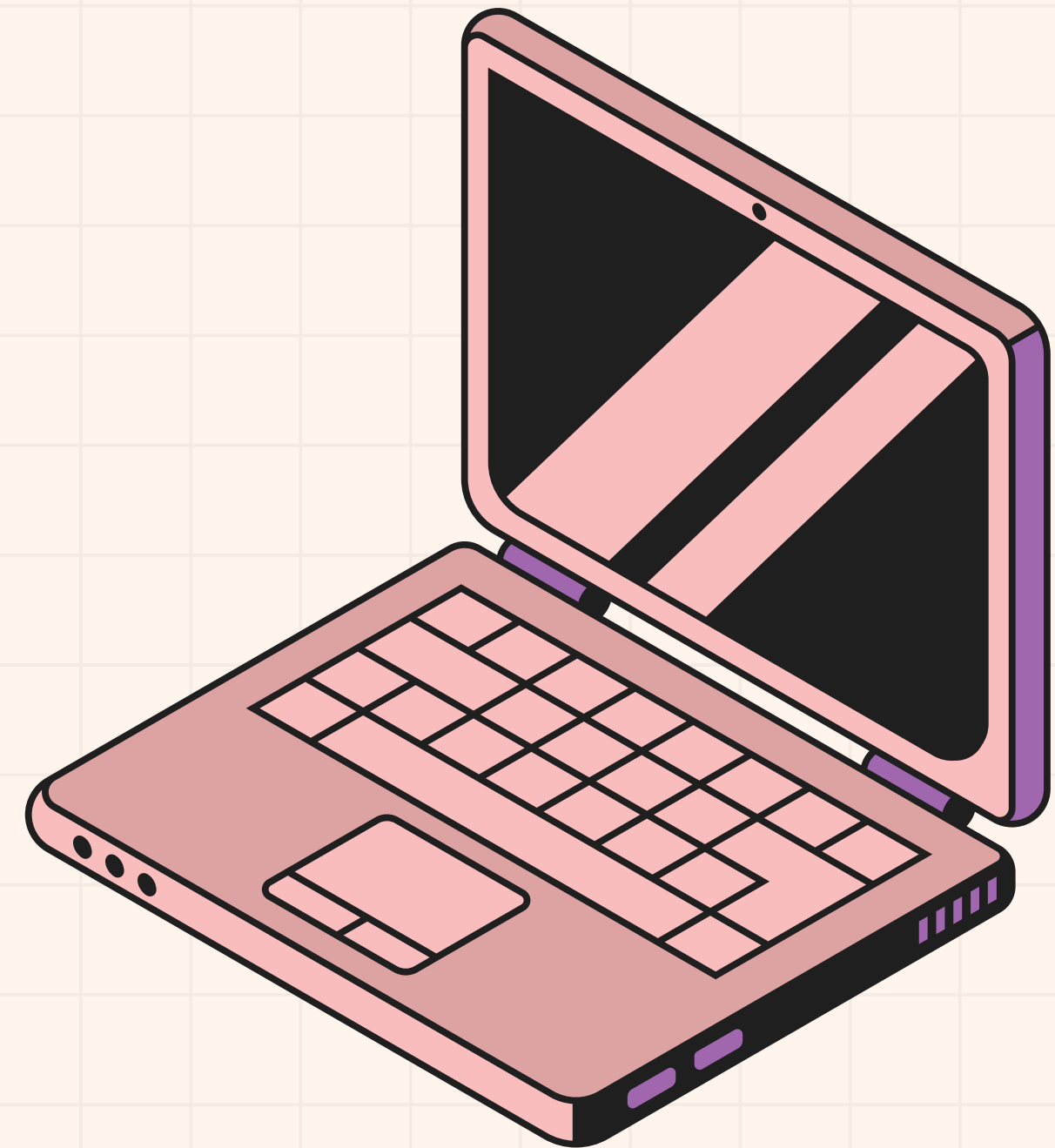
Um sistema distribuído completo para agendamento de serviços e gestão de clientes.

Cenário:

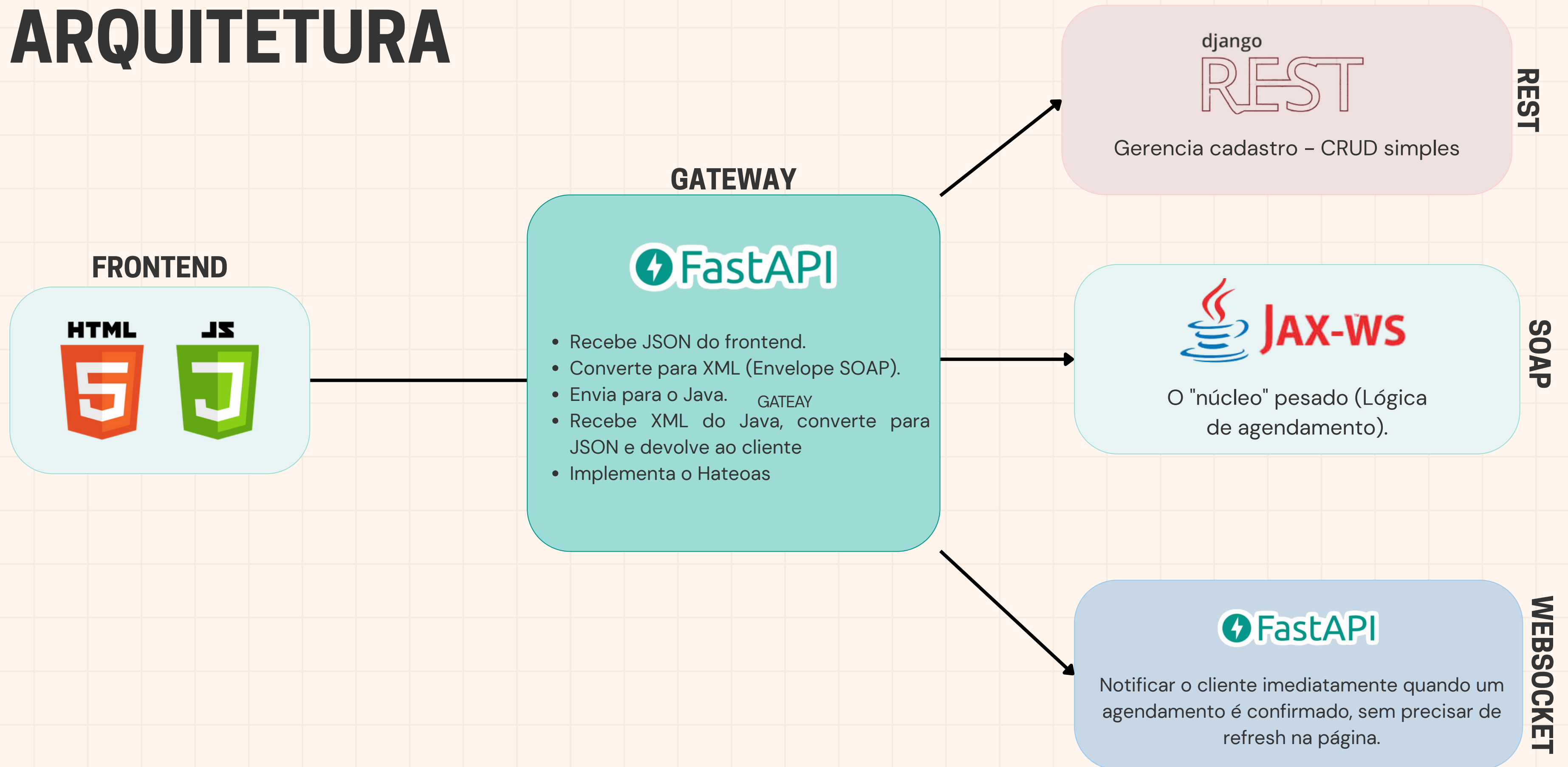
Simulamos um ecossistema de novas tecnologias (REST/Python) operando juntamente com sistemas robustos e legados (SOAP/Java).

Funcionalidades Principais:

- Cadastro de serviços e clientes.
- Verificação de disponibilidade e agendamento.
- Notificações instantâneas de confirmação.



ARQUITETURA



CONTRATO SOAP (JAVA)

```
package com.agendaja.soap;

import javax.jws.WebMethod;
import javax.jws.WebService;

// Interface do serviço de agendamento
@WebService
public interface AgendamentoService {

    @WebMethod
    String consultarDisponibilidade(String data);

    @WebMethod
    String agendarServico(int clienteId, int servicoId, String data, String horaInicio);

    @WebMethod
    String cancelarAgendamento(int id);

    @WebMethod
    String listarAgendamentos();
}
```

A INTEGRAÇÃO NO GATEWAY

```
# -----  
# CONFIGURAÇÕES DO SISTEMA  
# -----  
REST_URL = "http://localhost:5001"    # Django REST  
SOAP_WSDL = "http://localhost:8088/soap/agendamento?wsdl"  
  
soap_client = Client(SOAP_WSDL)  
  
@app.post("/agendar")  
def agendar(clienteId: int, servicoId: int, data: str, horaInicio: str):  
    resposta = soap_client.service.agendarServico(  
        clienteId, servicoId, data, horaInicio  
    )  
  
    # Envia notificação WebSocket  
    # for ws in connected_websockets:  
    #     ws.send_text(f"Novo agendamento: {data} às {horaInicio}")  
  
    return {"mensagem": resposta}
```

HATEOAS(NAVEGABILIDADE)



```
@app.get("/")
def gateway_root():
    return {
        "message": "API Gateway funcionando",
        "_links": {
            "self": "/",
            "servicos": "/servicos",
            "clientes": "/clientes",
            "agendar": "/agendar",
            "disponibilidade": "/disponibilidade?data=YYYY-MM-DD",
            "cancelar": "/cancelar",
            "listarAgendamentos": "/listarAgendamentos",
            #"websocket": "/ws",
        }
    }
```

WEBSOCKET



```
connected_websockets = []

@app.websocket("/ws")
async def websocket_endpoint(websocket: WebSocket):
    await websocket.accept()
    connected_websockets.append(websocket)

    try:
        while True:
            await websocket.receive_text()
    except:
        connected_websockets.remove(websocket)
```


CAMADA DE DADOS E PERSISTÊNCIA

django
REST



DJANGO ORM COM SQL LITE

 JAX-WS



JDBC PURO COM SQL NATIVO

OBRIGADA!

