Tornando o projeto assíncrono

> https://fastapidozero.dunossauro.com/4.0/08/

Objetivos da Aula

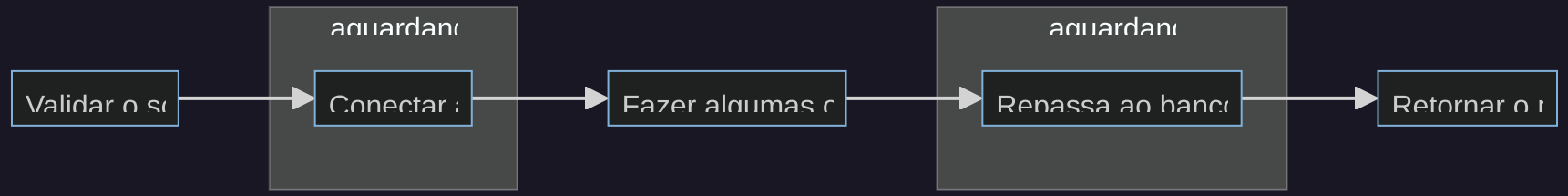
- Introduzir os conceitos de programação assíncrona
- Refatorar nossa aplicação para suportar AsyncIO
  - Tanto a aplicação
  - Quanto os testes

Uma pequena introdução ao AsyncIO

> Parte 1

Bloqueio de I/O

Um dos problemas mais comuns ao lidarmos com uma aplicação web é a necessidade de estarmos sempre disponíveis para responder a requisições enviadas pelos clientes. Quando recebemos uma requisição, normalmente precisamos processá-la:



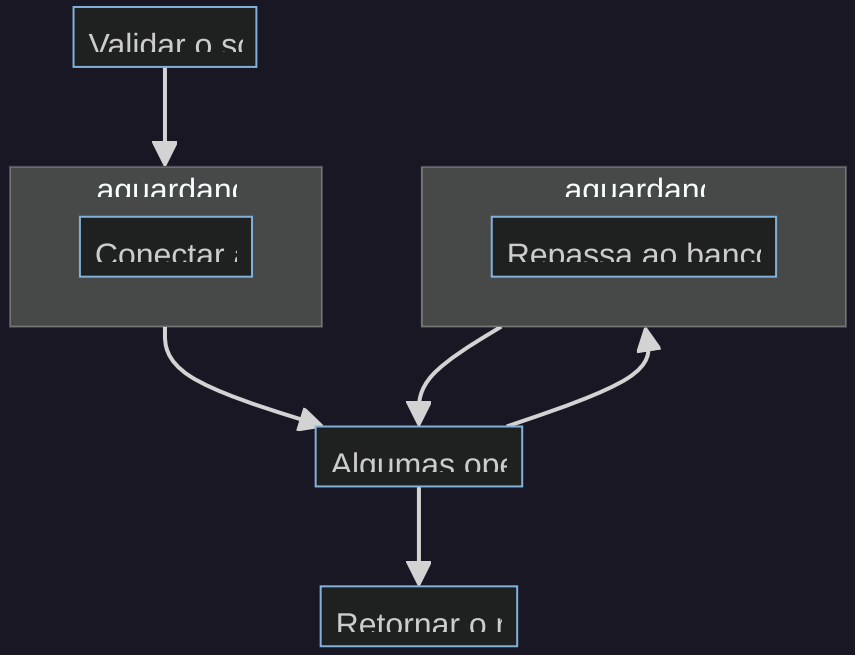
Esse fluxo de execução é chamado de **bloqueante**, pois a aplicação fica "parada", aguardando a resposta de sistemas externos, como o banco de dados.

Bloqueio de I/O

```python
@router.put('/{user_id}', response_model=UserPublic)
def update_user(
    user_id: int,
    user: UserSchema,
    session: Session,
    current_user: CurrentUser,
):
    if current_user.id != user_id:
        raise HTTPException(
            status_code=HTTPStatus.FORBIDDEN,
                        detail='Not enough permissions'
        )
    try:
        current_user.username = user.username
        current_user.password = get_password_hash(user.password)
        current_user.email = user.email
        session.commit()
        session.refresh(current_user)

        return current_user

    except IntegrityError:
        raise HTTPException(
            status_code=HTTPStatus.CONFLICT,
            detail='Username or Email already exists',
        )
```



Loop de eventos

Cooperatividade e Escalonamento