### **Relatório sobre Programação em Par no Projeto Biblox**

### **1. Organização Geral do Trabalho**

A estrutura do projeto foi modularizada em três partes principais: biblox (configurações e integração central), livros (gestão de livros) e usuários (gestão de usuários).  
 O trabalho foi distribuído em pares para garantir eficiência, qualidade do código e aprendizado compartilhado. Abaixo, estão listadas as principais atividades associadas à programação em par do projeto.

### **2. Etapas do Desenvolvimento e Contribuições das Duplas**

#### **2.1. Configuração Inicial do Projeto**

* **Dupla :** Marco e João Lucas (piloto: João Lucas, navegador: Marco)  
   Trabalharam na configuração inicial do projeto Django, definindo o ambiente base de desenvolvimento.  
   **Atividades realizadas:**
  + **Configuração do settings.py:**
    - Ajustaram o banco de dados para SQLite, habilitando as configurações padrão do Django.
    - Configuraram INSTALLED\_APPS para incluir as aplicações livros e usuários.
  + **Configuração do arquivo urls.py para conectar rotas principais de cada aplicação.**

**Destaque da colaboração:**

* + **Enquanto João Lucas implementava as configurações, Marco revisava recursos de segurança, como o uso de variáveis de ambiente.**

#### **2.2. Desenvolvimento do CRUD de Livros**

* **Dupla 2:** João Cláudio e Luís (piloto: João Cláudio, navegador: Luís)  
   Focados no módulo responsável pela manipulação dos livros, implementaram as funcionalidades principais de CRUD.  
   **Atividades realizadas:**
  + **Criação do modelo Livro:**
    - Implementaram atributos como título, autor, isbn e publicação.
    - Adicionaram validações para evitar duplicação de ISBN.
    - Relacionaram livros com usuários.
  + **Criação das views:**
    - João Cláudio criou funções para listar\_livros, detalhar\_livro e cadastro\_livro.
    - Luís sugeriu melhorias no desempenho usando querysets otimizados e revisou o código.
  + **Implementação e ligação das rotas:**
    - Criaram e conectaram as rotas para cada funcionalidade no arquivo urls.py.

#### **2.3. Desenvolvimento da Interface para Livros**

* **Dupla 3:** Juliana e Vitória (piloto: Juliana, navegador: Vitória)  
   Implementaram templates no Django para o módulo livros, priorizando a usabilidade do sistema. **Atividades realizadas:**
  + Criação do diretório templates/livros/ e arquivos HTML.
  + Desenvolveram views baseadas em templates para renderizar páginas.
    - Juliana implementou as páginas de listagem (listar\_livros.html) e cadastro (form\_livro.html).
    - Vitória revisou os templates, sugerindo a inclusão de mensagens de feedback e ajuste no layout com Bootstrap.

**Destaque da colaboração:**  
 Juliana e Vitória usaram Django Template Tags para criar uma interface simples e funcional.

#### **2.4. Implementação do Sistema de Usuários**

* **Dupla 4:** João Lucas e Luana (piloto: João Lucas, navegador: Luana)  
   Trabalharam nas funcionalidades de autenticação e cadastro de usuários. **Atividades realizadas:**
  + **Customização do modelo de usuário:**
    - Aderiram ao modelo padrão do Django com extensões adicionais (ex.: CPF e telefone).
  + **Criação de formulários personalizados para login e cadastro:**
    - João Lucas escreveu o formulário (baseado em UserCreationForm).
    - Luana revisou e adicionou validação extra para o CPF.
  + **Implementação das views e templates:**
    - Criaram páginas como login.html e cadastro.html, conectadas às views baseadas em classe do Django (LoginView, LogoutView).

#### **2.5. Integração e Testes**

* **Dupla 5:** Marco e João Lucas (piloto: Marco, navegador: João Lucas)  
   Foram responsáveis por integrar todas as partes do projeto e testar o sistema. **Atividades realizadas:**
  + **Testes unitários e de integração:**
    - **Usaram pytest para verificar:**
      * Validação do modelo.
      * Funcionamento das views de livros e autenticação.
  + **Feedback para ajustes:**
    - Marco e João Lucas ofereceram sugestões para corrigir problemas de validação e melhorar a experiência do usuário.

### **3. Benefícios Observados na Programação em Par**

1. **Qualidade de Código Melhorada:**
   * As revisões constantes durante a programação em par reduziram erros e melhoraram a legibilidade do código.
2. **Inovação e Colaboração:**
   * Os navegadores frequentemente propuseram soluções criativas para problemas, como otimizações em queries e melhoria na validação.
3. **Aceleração no Desenvolvimento:**
   * A divisão das tarefas em duplas permitiu trabalho paralelo em diferentes partes do sistema, sem comprometer a integração.
4. **Aprimoramento de Habilidades:**
   * Os programadores aprenderam uns com os outros, especialmente em áreas em que tinham menos experiência.

### **4. Desafios Enfrentados**

1. **Diferenças de Estilo:** Alguns momentos exigiram mais tempo para alinhar padrões de codificação.
2. **Velocidade Inicial Reduzida:** A programação em par pode parecer mais lenta no início, mas traz retornos a médio/longo prazo.
3. **Resolução de Conflitos:** Durante a integração, surgiram conflitos no Git que precisaram ser resolvidos manualmente.

### **5. Conclusão**

**A aplicação da programação em par no projeto Biblox trouxe claros benefícios, incluindo redução de falhas, aprendizado contínuo e um código mais limpo e extensível. A divisão de responsabilidades entre as duplas foi eficaz, mas alguns desafios (como alinhamento de estilos e resolução de conflitos) reforçam a importância de um planejamento contínuo.**