

## **Conversa Copilot (GPT 5.2)**

AUTOR:

prompt inicial

IA:

Antes de eu começar a criar o projeto e escrever o código, preciso alinhar alguns pontos que estão ambíguos (principalmente por envolver o "User" vs autenticação nativa do Django e o fluxo de status). Responda por favor:

1. Autenticação (Django nativo)

- Você quer login usando email (sem username) ou pode ser username padrão do Django (mantendo email/cnpj como campos extras)?
- Pode ser "criar usuário via admin" ou você quer tela de cadastro (signup) também?

2. Modelo User

- O campo name deve ser um campo único (ex: name) ou posso mapear para first\_name/last\_name do Django e expor como "Nome" na UI?
- cnpj precisa de validação de formato/dígitos (ex.: 14 dígitos com validação) ou apenas string obrigatória?

3. Fluxo de status (Production / ProductionMachine)

- Ao criar uma produção e associar máquinas,
  - status inicial deve ser:
    - Production: STANDBY ou ONGOING?
    - ProductionMachine: STANDBY ou ONGOING?

- Quando uma produção é marcada como FINISHED, o que deve acontecer com as máquinas ainda em HALT?
  - A regra permite finalizar se não houver STANDBY/ONGOING, então HALT “conta como pronto”; você quer permitir finalizar mesmo com HALT?
- O status HALT deve existir só no ProductionMachine: haverá ação na UI para “parar” (HALT) ou ele só existirá para representar algum estado interno (sem botão)?

#### 4. CRUD esperado

- Para máquinas e produções: além de cadastrar/listar, você quer editar também, ou pode ser só cadastro + listagem + ações (cancelar/finalizar)?
- “Soft delete”: você quer botão “Excluir” (marcando deleted\_at) para máquinas e/ou produções, ou não precisa expor isso na UI? Assim que você responder, eu implemento tudo de ponta a ponta (Django + templates HTML/CSS/JS + regras + Docker/Docker Compose + README multi-plataforma) direto no workspace.

Autor:

1. Padrão do Django; Tela de cadastro;
2. Deve ser único; Não precisa de validação
3. STANDBY em ambos; Sim, pode finalizar
4. Só cadastro e listagem; Precisa de botão na UI

IA:

1. Padrão do Django; Tela de cadastro;
2. Deve ser único; Não precisa de validação
3. STANDBY em ambos; Sim, pode finalizar
4. Só cadastro e listagem; Precisa de botão na UI

✓ Designed user and machine models

Vou primeiro inspecionar o workspace e inicializar a estrutura Django (projeto + apps), depois implemento models/regras/views/templates e por fim Docker/README.

✓ Configured Python environment

⚠ Create a Virtual Environment (3.12.3)?

Virtual Environments provide the benefit of package isolation and more.

Allow | ▾

Skip

✓ Configured Python environment

✓ Creating a Virtual Environment

Agora vou criar um plano curto e começar scaffold do Django + dependências.

✓ Created 5 todos

📦 Install Django==5.1.4 in Python 3.12.3

(.venv)?

Allow



Skip

Autor: allow all commands in this session

Tempo até terminar o projeto: ~7 min

Autor:

Erro ao executar docker-compose up --build

Attaching to crud-fabrica-web

Error response from daemon: failed to create task for container: failed to create shim task:  
OCI runtime create failed: runc create failed:  
unable to start container process: error during  
container init: exec: "/app/entrypoint.sh":  
permission denied: unknown

ⓘ README.md:49

Vou ajustar o entrypoint no container para rodar via  
`sh` (assim não depende do bit executável quando o  
volume `./app` sobrescreve o arquivo), e você  
poderá rodar `docker compose up --build` sem esse  
erro.

✓ Edited 🛡 Dockerfile +1 -3

```
● cd
  /home/berbeling/codes/pr
  ojeto/copilot && docker
  compose up --build
```

Corrigido: o erro vinha do bind mount `./app`  
sobrescrevendo o `entrypoint.sh` e removendo o  
bit executável, então o Docker tentava executar  
`entrypoint.sh` e recebia "permission denied".

- Ajustei o 🛡 Dockerfile para usar `ENTRYPOINT`  
`["sh", [entrypoint.sh]]` (`http://_vscodecontentref_/_3)` (não  
depende de permissão de execução).
- Validei subindo com `docker compose up --`  
`build`: o container iniciou, aplicou migrations e o  
Django subiu em `http://0.0.0.0:8000/`.

Pode rodar agora:

- `docker compose up --build`

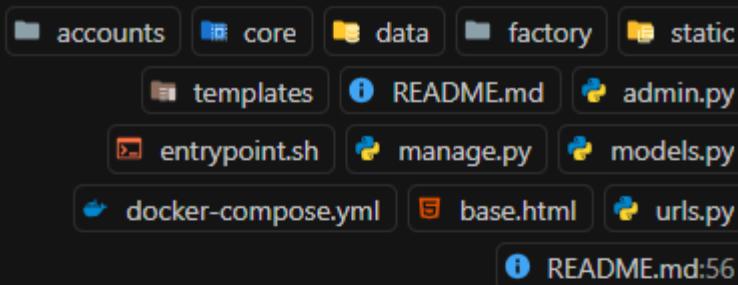
AUTOR:

prompt incremental

IA:

Responda de forma técnica, clara e objetiva.

Nunca abrevie ou resuma trechos de código em suas respostas.



- ✓ Evaluated production statuses and updated access

Vou implementar: `is_premium` (com controle no admin), ajuste do limite de máquinas (5/10), `working_time` automático em

`ProductionMachine`, toggle Light/Dark com persistência, e exibição de `working_time` na lista de produções; depois gero migrations e valido com `manage.py check`.

- ✓ Created 5 todos
- ✓ Edited `models.py +1 -0`
- ✓ Edited `admin_forms.py +12 -1`
- ✓ Edited `admin.py +26 -3`
- ✓ Edited `forms.py +3 -2`
- ✓ Edited `machines.html +1 -1`

Agora vou implementar `working_time` com cálculo automático (incluindo cancelamento em massa da produção) e, em seguida, o toggle Light/Dark + exibição do `working_time` na lista de produções.

- ✓ Edited `models.py +31 -7`
- ✓ Edited `views.py +6 -0`
- ✓ Edited `base.html +31 -0`
- ✓ Edited `app.css +14 -0`
- ✓ Edited `views.py +5 -1`
- ✓ Edited `productions.html +15 -0`



- ✓ Summarized conversation history
- ✓ Planned migrations and ran validation checks

Vou gerar as migrations novas (accounts/factory),  
aplicar no SQLite e rodar `check /tests` para garantir  
que nada quebrou.

- ✓ Configuring a Python Environment

```
● cd /home/berbeling/codes/projeto/copilot && /home/berbeling/codes/projeto/copilot/.venv/bin/python manage.py makemigrations accounts factory
```

```
● cd /home/berbeling/codes/projeto/copilot && /home/berbeling/codes/projeto/copilot/.venv/bin/python manage.py migrate
```

```
● cd /home/berbeling/codes/projeto/copilot && /home/berbeling/codes/projeto/copilot/.venv/bin/python manage.py check
```

```
● cd /home/berbeling/codes/projeto/copilot && /home/berbeling/codes/projeto/copilot/.venv/bin/python manage.py test
```

- ✓ Managed todo items and updated README details
- ✓ Edited [README.md](#) +12 -2
- ✓ Checked migration files and updated README
- ✓ Edited [README.md](#) +10 -0

- ✓ Edited [README.md +10 -0](#)
- ✓ Checked migrations setup and crafted commands
  - Criei e apliquei as migrations pendentes:  
[0002\\_user\\_is\\_premium.py](#) e  
[0002\\_productionmachine\\_working\\_time.py](#).
  - Rodei validação: `python [manage.py]`  
([http://\\_vscodecontentref\\_/3](#)) `check` sem issues; `python [manage.py]`  
([http://\\_vscodecontentref\\_/4](#)) `test` não tem testes no projeto (0 tests).
  - Atualizei a documentação em [README.md](#) para refletir limite 5/10 (premium),  
[working\\_time](#) e o toggle Light/Dark, e incluí um passo de “atualizar em produção (Docker)”.

Se quiser, eu também posso adicionar testes básicos cobrindo o cálculo do [working\\_time](#) e as regras de finalização/cancelamento.

Tempo até terminar o projeto: ~3 min