Arquivo: setor_responsavel.py

```
from PyQt6.QtWidgets import QVBoxLayout, QHBoxLayout, QLabel, QLineEdit, QComboBox, QTextEdit

def create_dados_responsavel_contratacao_group(data):
    """Cria o layout para o responsável pela contratação usando programação funcional."""
    layout = QVBoxLayout()

    # Função auxiliar para criar um layout de linha com um label e um campo de entrada
    def create_info_row(label_text, widget):
        row_layout = QHBoxLayout()
        label = QLabel(label_text)
        row_layout.addWidget(label)
        row_layout.addWidget(widget)
        return row_layout

    # Par, Prioridade e CP
    par_edit = QLineEdit(str(data.get('cod_par', '')))
    prioridade_combo = QComboBox()
    prioridade_combo.addItems(["Necessário", "Urgente", "Desejável"])
    prioridade_combo.setCurrentText(data.get('prioridade_par', 'Necessário'))

    cp_edit = QLineEdit(data.get('cp', ''))

    par_layout = QHBoxLayout()
    par_layout.addLayout(create_info_row("Número da CP:", cp_edit))
    par_layout.addLayout(create_info_row("Meta do PAR:", par_edit))
    par_layout.addLayout(create_info_row("Prioridade:", prioridade_combo))
    layout.addLayout(par_layout)

    # Endereço
    endereco_edit = QLineEdit(data.get('endereco', ''))
    endereco_cep_layout = QHBoxLayout()
    endereco_cep_layout.addLayout(create_info_row("Endereço:", endereco_edit))
    layout.addLayout(endereco_cep_layout)

    # E-mail
    email_edit = QLineEdit(data.get('email', ''))
    email_layout = QHBoxLayout()
    email_layout.addLayout(create_info_row("E-mail:", email_edit))
    layout.addLayout(email_layout)

    # CEP e Telefone
    cep_telefone_layout = QHBoxLayout()
    cep_edit = QLineEdit(str(data.get('cep', '')))
    cep_telefone_layout.addLayout(create_info_row("CEP:", cep_edit))
    telefone_edit = QLineEdit(data.get('telefone', ''))
    cep_telefone_layout.addLayout(create_info_row("Telefone:", telefone_edit))
    layout.addLayout(cep_telefone_layout)

    # Dias e Horário para Recebimento
    dias_edit = QLineEdit(data.get("dias_recebimento", "Segunda à Sexta"))
```

```
    horario_edit = QLineEdit(data.get("horario_recebimento", "09 às 11h20 e 14 às 16h30"))

    layout.addLayout(create_info_row("Dias para Recebimento:", dias_edit))
    layout.addLayout(create_info_row("Horário para Recebimento:", horario_edit))

    # Justificativa
    justificativa_label = QLabel("Justificativa para a contratação:")
    justificativa_label.setStyleSheet("font-size: 12pt;")
    justificativa_edit = QTextEdit(data.get("justificativa", ""))
    layout.addWidget(justificativa_label)
    layout.addWidget(justificativa_edit)

    # Armazena widgets, incluindo os grupos
    widget_setor_responsavel = {
        'cp_edit': cp_edit,
        'par_edit': par_edit,
        'prioridade_combo': prioridade_combo,
        'endereco_edit': endereco_edit,
        'cep_edit': cep_edit,
        'email_edit': email_edit,
        'telefone_edit': telefone_edit,
        'dias_edit': dias_edit,
        'horario_edit': horario_edit,
        'justificativa_edit': justificativa_edit,
    }



    return layout, widget_setor_responsavel

def get_justification_text():
    pass
    # current_justification = self.df_registro_selecionado['justificativa'].iloc[0]

    # # Retorna o valor atual se ele existir, senão, constrói uma justificativa baseada no tipo de material/serviço
    # if current_justification:  # Checa se existe uma justificativa
    #     return current_justification
    # else:
    #     # Gera justificativa padrão com base no tipo de material ou serviço
    #     if self.material_servico == 'Material':
     #           return (f"A aquisição de {self.objeto} se faz necessária para o atendimento das necessidades do(a) {self.setor_responsavel} do(a) {self.orgao_responsavel} ({self.sigla_om}). A disponibilidade e a qualidade dos materiais são essenciais para garantir a continuidade das operações e a eficiência das atividades desempenhadas pelo(a) {self.setor_responsavel}.")
    #     elif self.material_servico == 'Serviço':
      #            return (f"A contratação de empresa especializada na prestação de serviços de {self.objeto} é imprescindível para o atendimento das necessidades do(a) {self.setor_responsavel} do(a) {self.orgao_responsavel} ({self.sigla_om}).")
    #     return ""
```