

Universidade Tecnológica Federal do Paraná



*Engenharia de Software II*

Felipe Veiga Ramos  
1061461

PROJETO 4 - PLANEJAMENTO DE  
DESENVOLVIMENTO DE SOFTWARE

Departamento de Ciência da Computação (DACOM)

# Conteúdo

# 1 Da Natureza Deste Documento

Este plano objetiva descrever e estimar custos e tempo para a criação do sistema CVLates.

O sistema CVlates objetiva a criação de um gerenciador de currículos acadêmicos.

Baseado no documento de requisitos e no laudo do PBR este documento deverá estimar o custo e tempo necessários para a elaboração, criação e implantação do sistema CVLates.

Criação de um gerenciador de currículos, semelhante ao Lates já existente.

## 2 Descrição de Atividades

Para permitir o melhor planejamento e otimização do tempo, bem como o melhor emprego de cada colaborador da equipe, tendo em vista que existe um colaborador por função, (analista do projeto, programador, testador e gerente), o projeto é fracionado em diversas etapas.

### 2.1 Etapa 1: Diagramação

Composta por atividades documentais que visam explicitar e clarificar os requisitos e funcionalidades que devem ser atendidos no desenvolvimento. Como artefato, deverá gerar os diagramas de classe e casos de uso e quaisquer outros que auxiliem na compreensão, se julgados necessários.

Esta etapa é destinada ao projetista do sistema. Ele terá cerca de 5.49 horas para a elaboração dos documentos.

#### 2.1.1 Diagramação: Casos de Uso – A

Diagramação dos casos de uso a ser realizada de acordo com o documento de requisitos. Espera-se que sejam necessárias 3,16 horas.

#### 2.1.2 Diagramação: Diagramas de Classe – B

Desenvolvimento do diagrama de classes: interface mais próxima da programação, necessária para nortear os trabalhos do programador. São esperadas 2,33 horas de trabalho.

### 2.2 Sequenciamento de atividades: Ordem de atividades do projetista

O diagrama de casos de uso deverá estar disponível antes de iniciar o desenvolvimento do diagrama de classes. Cada um cumpre uma importante função – enquanto o diagrama de casos de uso permite compreender as reais necessidades e requisitos, o de classe uma ideia de como elas serão implementadas.

Sequência de atividades até então:

Atividade	Atividade precedente	Duração Estimada (Horas)
A	-	3,16
B	A	2,33
Total:		5,49

Tabela 1: tabela de sequenciamento de atividades: Diagramação.

## 2.3 Etapa 2: Implementação

Aqui é descrito o sequenciamento de atividades a partir do momento em que se tem em mãos a diagramação completa dos requisitos funcionais a partir da documentação. Em outras palavras, nesta etapa o projetista já gerou documentos suficientes para que o programador interprete o que será implementado. A implementação será dividida em três etapas, sendo a primeira referente a implementação das classes que compõem o sistema e a segunda referente ao banco de dados que deverá suportar o armazenamento informacional. A última etapa é referente à interface gráfica e será tratada no final. O tempo total previsto para esta etapa é de 36,5 horas.

### 2.3.1 Implementação: Classes – C

Nesta etapa, de acordo com o que pode ser visto na previsão de consumo de tempo, o programador terá cerca de 15,5 horas para concluir a implementação de classes do sistema. Esta é necessariamente a primeira etapa do desenvolvimento do sistema, sendo que a modelagem e implementação do banco de dados será a etapa consecutiva a esta.

### 2.3.2 Implementação: Banco de Dados – D

Esta é a segunda etapa do desenvolvimento, sendo que para esta é previsto um consumo de tempo de aproximadamente 7,17 horas.

### 2.3.3 Implementação: Interface Gráfica – E

Última etapa do desenvolvimento, na qual está previsto o gasto de 13,83 horas de implementação.

## 2.4 Sequenciamento de Atividades: Ordem de atividades do Programador

O programador seguirá as etapas descritas anteriormente afim de manter a organização do processo de desenvolvimento parcial das partes do sistema. Estas serão distinguíveis entre si quando ao contexto de implementação, bem como, sua representatividade perante ao sistema como um todo. A tabela a seguir exibe o sequenciamento até então tido nas etapas de desenvolvimento.

Atividade	Atividade Prescedente	Duração Estimada (Horas)
A	-	3,16
B	A	4,5
C	A,B	15,5
D	C	7,17
E	C,D	13,83

002.png

Figura 1: Sequenciamento de Atividades: adicionadas as três etapas de implementação de código. Isto sendo C a implementação de Classes, D a implementação de Banco e E a de interface gráfica. Os números são as horas consumidas

## 2.5 Etapa 3: Realização de testes – F

Esta etapa é responsabilidade do *tester*. Ele deverá encontrar o maior número de erros possíveis. Esta etapa só será aplicada após o término das implementações massivas do sistema. Segundo a estimativa, é previsto que o responsável pela etapa consumirá cerca de 5,3 horas para desenvolver testes de performance, confiabilidade, integridades e dentre outros aspectos do sistema. Após este passo, o *relatará ao gerente* os erros encontrados no sistema para que este seja ao programador para correções.

### 2.5.1 Sequenciamento: Ordem de Atividades até a etapa do *tester*

Aqui verifica-se o sequenciamento do desenvolvimento até a etapa do *tester*

Atividade	Atividade Prescedente	Duração Estimada (Horas)
A	-	3,16
B	A	4,5
C	A,B	15,5
D	C	7,17
E	C,D	13,83
F	C,D,E	5,3

003.png

Figura 2: Sequenciamento de atividades: os números atrelados aos nós são relativos ao consumo em horas por atividade

## 2.6 Etapa 4: Correções de *bugs* e outros defeitos – G

Esta etapa é referente à correção de defeitos encontrados pelo *tester*. Assim, é um dever do programador realizar a manutenção. Por sua vez, este terá um tempo aproximadamente de 8,5 horas para conclusão. Com isto, encerram-se as etapas de produção e o gerente fará

uma avaliação final do produto (verificar se o produto tem qualidade mínima e atende os requisitos funcionais)

## 2.7 Sequenciamento: Da ordem de atividades até a etapa de correção pelo programador

Aqui verifica-se o desenvolvimento sequencial até a etapa de correção de *bugs*.

Atividade	Atividade Prescedente	Duração Estimada (Horas)
A	-	3,16
B	A	4,5
C	A,B	15,5
D	C	7,17
E	C,D	13,83
F	C,D,E	5,3
G	F	8,5

Tabela 2: Tabela de atividades que expressa todos os passos de desenvolvimento do projeto.

## 2.8 Etapa 5: Validação de Correções – H

Esta etapa compete ao gerente. Ele deverá assegurar que o sistema está pronto para o uso. A partir daqui está definido o término das atividades de desenvolvimento e o produto será liberado para o cliente. Devido a este fato, não convém uma estimativa de tempo ou assegurar espaço em cronograma, pois esta eventualidade é apenas um *checkpoint* do término e da entrega ao cliente.

## 3 Da Tabela de Atividades

As tabelas de atividade expressam o sequenciamento de tarefas ao longo do projeto. Entretanto, além delas é disponibilizado a estrutura de um grafo que exhibe desde o marco de início até o marco de entrega do produto (contendo as etapas intermediárias). Na seção a seguir, uma especificação maior do que foi elaborado neste planejamento de desenvolvimento de software.

A seguir, a tabela de atividades (porém com um detalhamento maior):

Percebe-se que o tempo total estimado em horas para a execução do projeto é de 57,96 horas. Isto é apenas uma estimativa da quantia total. A seguir o grafo sequencial de tarefas e posteriormente a elaboração do cronograma de desenvolvimento incluindo datas estipuladas para conclusões de etapas de desenvolvimento (Do início ao término – todas as etapas)

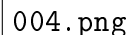
Atividade	Descrição	Atividade Precedente	Duração Estimada (Horas)
A	D. Casos de Uso	-	3,16
B	D. de Classes	A	4,5
C	I. Classes	A,B	15,5
D	I. BD	C	7,17
E	I. interface	C,D	13,83
F	Testes	C,D,E	5,3
G	C. de Defeitos	F	8,5
H	Validação de Software	G	–

Tabela 3: quadro detalhado do sequenciamento de atividades

## 4 Grafo determinístico: Sequenciamento estratégico de atividades

Aqui está explicitado a versão completa da estrutura que define a sequência de atividades do projeto. Tal não foi definida com o critério do custo de menor tempo (utilização do conceito de caminho crítico), mas sim com base no aspecto de desenvolvimento incremental e do uso de **MVC** – *Model, View, Controller*. Assim, o critério de construção da sequência de atividades pode ser entendido como um grafo que define um caminho de dependências do projeto (leva-se em consideração que se tem disponível apenas um programador)

A seguir, o grafo explicitado (os pesos estão rotulados em seus nós, e não em arestas – meramente ilustrado para compreensão da sequência de etapas):



004.png

## 5 Cronograma de atividades

No cronograma de atividades não será especificado qual horário ou dia o participante do projeto deverá desenvolver. Entretanto, como é necessário ter um controle do processo de desenvolvimento para que este siga prazos e venha a fazer parte de um rol de metas é necessário estipular prazos. A única obrigação referente a datas é a entrega do trabalho no tempo correto e juntamente entregar as marcações de dias e horas trabalhadas. Isto para verificação a respeito das estimativas e da eficiência do planejamento realizado.

Segue o cronograma:

1. Desenvolvimento de Casos de Uso: Entrega em 17/06/16
2. Desenvolvimento de Diagrama de Classe: Entrega em 17/06/16
3. Implementação de Classes: Entrega em 24/06/16
4. Implementação de Banco de Dados: Entrega em 26/06/16
5. Implementação de Interface Gráfica: 30/06/16

6. Realização de testes: Entrega 02/07/16
7. Correção de defeitos: Entrega 05/07/16
8. Avaliação do produto e entrega: 06/07/16

## **5.1 Justificativas do Cronograma de atividades**

Como o processo de desenvolvimento é simplificado quanto as atividades concomitantes, pelo fato de que as atividades possuem dependência entre si e a forma de desenvolvimento é similar às metodologias tradicionais, só há um caminho na rede. Assim, não foi necessário utilizar métodos complexos para calcular qual o melhor caminho a ser tomado na decisão da sequência de atividades.

Isto define que este caminho único seja o caminho crítico. A respeito das datas, é tido como ponto de partida o dia de hoje (15/06/16), dia em que este documento foi lançado. Assim, a primeira etapa corrente será o desenvolvimento de diagrama de casos de uso. A respeito da quantidade de dias, foi tomado a decisão de dar um tempo extra, considerando possíveis fatores humanos que os desenvolvedores possuem rotinas fora do projeto. Assim, o planejamento tem um cronograma que totaliza 22 dias.

## **6 Plano de Contenção: Possíveis problemas do desenvolvimento**

Aqui considera-se que fatores humanos podem intervir no plano de desenvolvimento e possivelmente vir a decorrer na falha da execução das previsões de consumo de tempo, bem como, nas entregas nas datas previstas. Desse modo, cabe ao gerente estar ciente da situação dos membros da equipe e assumir a responsabilidade por elementos não concluídos ou falhos ao término do projeto.

Por causa disso, é necessário salientar que o gerente intervirá em etapas do processo de desenvolvimento quando este analisar a deficiência na execução do plano de projeto.

## **7 Metodologia empregada: Como foi realizado este planejamento de desenvolvimento de software**

Este planejamento de software foi realizado com base em dois fatores de escolha organizacional: metodologia de desenvolvimento tradicional e padrão arquitetural de software adotado. Tudo será elaborado de forma incremental e modular, sendo que uma etapa anterior sempre será subsídio da próxima etapa (principalmente quanto a implementação) e estas definem as atividades de software.