

Projeto AS: plano de execução [draft]

Atualizado em 2022-04-05. Sujeito a revisão ([usar sempre a versão online](#) deste documento).

1 Enquadramento do projeto de grupo	1
1.1 Objetivos.....	1
1.2 Critérios de avaliação	2
2 Plano e atividades	2
3 Conteúdos dos entregáveis	2
3.1 E1: <i>Backlog</i> do projeto	2
3.2 E2: Relatório - Visão do projeto	3
3.3 E3 Relatório de análise	3
3.4 E5 Relatório técnico dos incrementos (desenvolvimento)	3
3.4.1 E5-1 Iteração #3	4
3.4.2 E3-2 Iteração #4	4
3.5 E4: Protótipo exploratório	4
3.6 E6: Implementação	4
4 Conteúdo indicativo das Apresentações	5
4.1 Apresentação 1: conceito do produto (=visão)	5
4.2 Apresentação 2: antevisão dos cenários suportados no novo sistema	5
4.3 Apresentação 3: validação a arquitetura	5
4.4 Apresentação 4: <i>user stories</i> implementadas e perspectiva geral do projeto	5

1 Enquadramento do projeto de grupo

1.1 Objetivos

Com o trabalho de projeto, pretende-se que os alunos desenvolvam uma especificação para um sistema de informação, aplicando um **método** de análise de sistemas e **modelos** visuais para caraterizar diferentes partes do SI.

No final, o resultado deve incluir:

- Uma proposta de transformação digital de uma área de atividade/negócio;
- A especificação dos novos processos de trabalho/atividades suportados no SI;
- A especificação de requisitos funcionais com modelos de casos de utilização;
- Especificação de atributos de qualidade;
- Modelo do domínio (mapa de conceitos);
- Protótipos exploratórios como estratégia de validação de requisitos.
- Implementação exploratória (camada de apresentação, na Web).

O trabalho é feito em **grupo**, funcionando as equipas definidas para as Práticas.

1.2 Critérios de avaliação

[t.b.c.]

2 Plano e atividades

O trabalho do projeto deve ser desenvolvido através de iterações, cada qual com um foco particular.

Cada iteração cria e/ou atualiza um conjunto de resultados. **No final de cada iteração há lugar à entrega dos resultados indicados.** A entrega é feita no eLearning. Conferir, no link de submissão, eventuais detalhes quanto ao formato da submissão ou ajuste de prazos.

Iter. #	Foco	Resultados/entregáveis
#1 02/05 09/05	Desenvolvimento da visão/conceito do projeto. Preparação das ferramentas e ambiente de trabalho.	<ul style="list-style-type: none">E1: Sistema para a gestão do <i>backlog</i>.E2: Relatório Técnico: VisãoApresentação 1 (na aula P da 2ª semana)
#2 16/05 23/05	Análise dos casos de uso. Modelo do domínio. Prototipagem das interações.	<ul style="list-style-type: none">Atualização do E1.E3: Relatório de AnáliseE4: Protótipo exploratório.Apresentação 2 (na aula P da 2ª semana)
#3 30/05 06/06	Validação da arquitetura candidata. Implementação exploratória.	<ul style="list-style-type: none">Atualização do E1.E5.1: Relatório Técnico (iteração 3)E6.1: Implementação exploratória.Apresentação 3 (na aula P da 2ª semana)
#4 13/06 20/06	Implementação das histórias “core”.	<ul style="list-style-type: none">Atualização do E1.E5.2: Relatório Técnico (iteração 4)E6.2: Implementação (histórias core).Apresentação 4 (na aula P da 2ª semana).

3 Conteúdos dos entregáveis

Todos os produtos (*outputs* do processo de especificação) **podem ser atualizados** de modo a refletir o progresso no projeto, a nova compreensão do problema ou, simplesmente, a mudança de ideias da equipa. Os relatórios incluem uma secção inicial de controlo de versões que deve ser atualizada em conformidade.

3.1 E1: *Backlog* do projeto

O *backlog* é um ambiente de gestão do projeto, usado para fazer o planeamento das tarefas atividades e para monitorar o progresso da construção do produto.

O *backlog* deve ser operacionalizado numa ferramenta online, que suporte os conceitos do planeamento ágil, à maneira da SCRUM; deve suportar os conceitos de iterações/*sprint*, *user stories* e *workflow*. E.g.: [JIRA](#), [GitHub+ZenHub](#), [PivotalTracker](#).

Algumas práticas a observar:

- todos os membros da equipa podem editar o *backlog*;
- *product owner* (o docente) aceita/rejeita o trabalho entregue.
- divisão do projeto em iterações e atribuição de “tarefas” às iterações. Todo o trabalho tem de ser planeado e priorizado (no *backlog*) antecipadamente.

As entradas predominantes do *backlog* devem ser *user stories*. As *user stories* representam “fatias” dos cenários dos casos de utilização, através de exemplos para uma forma de o caso de utilização acontecer. As *user stories* recebem uma estimativa de esforço, numa escala relativa (*story points*).

Os projetos incluem também itens de trabalho que, apesar de necessários, não correspondem diretamente a uma funcionalidade no produto (e.g.: preparar um relatório).

The screenshot shows a JIRA user story card. At the top, the title is 'O Joaquin cria um novo trabalho para casa'. Below the title, there's a 'Start' button and a dropdown menu set to 'Unscheduled'. A 'Collapse' button is also visible. The card displays various fields: 'ID' is #180704671, 'STORY TYPE' is Feature, 'PRIORITY' is P1 - High, 'POINTS' is 2 Points, 'REQUESTER' is Ilidio Oliveira, 'OWNERS' is <none>, and 'FOLLOW THIS STORY' is checked with 1 follower. A 'BLOCKERS' section has a '+ Add blocker or impediment' button. The 'DESCRIPTION' section contains the text: 'Sendo o docente Joaquin quero criar um novo trabalho para casa para os alunos de modo a que os alunos consolidem os conhecimento e fiquem inteirados do trabalho pendente. Critérios de aceitação: Cenário 1: trabalho com submissão de ficheiros criado com sucesso Quando xxx'. At the bottom, there's a 'LABELS' section with a tag 'implementação'.

User stories e implementação

Para a implementação dos casos de utilização nucleares a equipa deve identificar as funcionalidades pretendidas do produto, e apresentá-las no *backlog*. As *stories* da implementação devem ter uma pontuação e critérios de aceitação definidos, e podem usar personagens (persona).

3.2 E2: Relatório - Visão do projeto

O documento de Visão do Projeto é o relatório técnico que sai da Iteração #0; responde à pergunta: qual é a proposta de valor e para quem?

Apresenta a oportunidade que sustenta o desenvolvimento do projeto e, de forma geral, o âmbito das funcionalidades esperadas. O objetivo é suportar a decisão de avançar/não-avançar com o projeto (sem entrar em detalhes das funcionalidades).

O OpenUP fornece um [template](#) e [recomendações](#) para a sua preparação.

Para o projeto de MAS, o modelo a utilizar deve ser este:

→ [modelo para o relatório de Visão \[recurso do projeto\]](#)

3.3 E3 Relatório de análise

[t.b.c.]

3.4 E5 Relatório técnico dos incrementos (desenvolvimento)

3.4.1 E5-1 Iteração #3

[t.b.c.]

3.4.2 E3-2 Iteração #4

[t.b.c.]

3.5 E4: Protótipo exploratório

O protótipo exploratório apresenta, num ambiente interativo, os cenários de utilização a suportar no sistema. É um demonstrador do sistema, sem ainda haver sistema implementado, ou seja, mostra uma previsão de como é que os produtos de software (*sites*, aplicações móveis,...) irão funcionar quando forem construídos.

O protótipo exploratório deve ser suficientemente **elaborado** e realista para:

- ajudar a equipa a comunicar o conceito com clareza e objetividade;
- ajudar o cliente/stakeholders a validar os requisitos.
- cativar os stakeholders.

... **mas** suficientemente **simples** para:

- ser construído rapidamente, já que a natureza exploratória significa que se pode estar a considerar cenários que podem ser abandonados.
- não tem código (implementação) nem base de dados (a informação é demonstrativa e estática).

Uma vez que se pretende antecipar num ambiente exploratório a experiência de utilização futura, o protótipo deve apresentar **dados credíveis**, compatíveis com aquilo que iria aparecer numa versão de produção.

O protótipo pode ser construído com a ajuda de ferramentas de prototipagem rápida (e deve permitir “navegar” entre páginas/ecrãs do protótipo). Alguns ambientes sugeridos: [Figma](#), [VisualParadigm](#) (UX design); <https://proto.io/>, <http://www.invisionapp.com/>

3.6 E6: Implementação

A implementação solicitada no âmbito é a construção parcial da camada de apresentação, num ambiente Web (ou *mobile*, se aplicável).

Os grupos devem priorizar os casos de utilização e identificar aqueles que representam a funcionalidade central da solução. Estes casos de utilização devem ser implementados.

O grau de sofisticação da implementação pode variar consoante os conhecimentos já adquiridos pelos elementos da equipa, mas, no mínimo, consiste num conjunto de páginas, **instaladas** num servidor, em que se pode experimentar o fluxo principal da aplicação.

Para esta implementação, deve ser usado um ambiente de programação e não um ambiente de prototipagem rápida. Os alunos podem, naturalmente, utilizar as tecnologias aprendidas em ITW ou outras que dominam.

A implementação deve ser coerente com os modelos (e.g.: reflete os cenários dos casos de utilização, utiliza os conceitos definidos no modelo do domínio).

A camada de acesso a dados pode ser implementada com uma solução local; não é pedido (nem esperado) o uso de bases de dados.

E6-1 Implementação Inicial

Pretende-se uma primeira abordagem, **muito simples**, mas que mostre já a utilização de um projeto de código/desenvolvimento, com as tecnologias selecionadas.

E6-2 Implementação dos casos de utilização nucleares

Implementação simples, que permita demonstrar a utilização dos **casos de utilização nucleares**. A solução deve estar alojada num servidor web ¹.

4 Conteúdo indicativo das Apresentações

Estão previstas as seguintes apresentações em que a equipa de “analistas” comunica as ideias para o novo sistema:

- Apresentação 1: Conceito do produto
- Apresentação 2: Previsão dos cenários a suportar (protótipo)
- Apresentação 3: Validação da arquitetura
- Apresentação 4: Apresentação final dos resultados e demonstração dos casos de utilização nucleares.

O conteúdo esperado para cada apresentação é a seguir detalhado.

4.1 Apresentação 1: conceito do produto (=visão)

[t.b.c.]

4.2 Apresentação 2: antevisão dos cenários suportados no novo sistema

[t.b.c.]

4.3 Apresentação 3: validação da arquitetura

[t.b.c.]

4.4 Apresentação 4: *user stories* implementadas e perspetiva geral do projeto

[t.b.c.]

¹ A aplicação deve ser colocada num servidor de modo a ser acessível através de um URL (*hosted*). Notar que existem alguns servidores gratuitos disponíveis para alojar *sites* simples (que não precisam de *frameworks*), e.g.: [áreas pessoais na UA](#); [GitHub pages](#). Para soluções mais complexas, existem ainda soluções com “*free tier*” como o [Heroku](#) ou o [Firebase](#).