 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático II	Ano letivo 2021/2022	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software II	Duração	

Observações

- Respeite as datas pré-definidas para entrega dos resultados do trabalho prático.
- Toda a componente de desenvolvimento deverá ser em linguagem Java.
- A submissão dos resultados do trabalho deverá ser efetuada através do moodle em <http://moodle.estg.ipp.pt>

Os docentes: Cristóvão Sousa, João Carneiro, Marco Gomes e Óscar Oliveira

Enunciado

O presente enunciado apresenta um desafio no domínio da engenharia de software, em particular em atividades que assegurem a qualidade de software. Concretamente, procura explorar os conceitos demonstrados na unidade curricular sobre metodologias de teste de software, metodologias de desenvolvimento ágeis, *Continuous Integration* (CI) e planos de *Software Configuration Management* (SCM).

Descrição do problema

A Market for Business to Business (G4B²) é uma plataforma que permite facilitar a transação de produtos entre empresas a nível nacional. A empresa, para além de ser permitir efetuar as transações, faz a recolha e entrega das encomendas. A empresa tem um armazém em cada distrito e uma frota de veículos para recolha e entrega de encomendas. Para melhorar a produtividade da G4B², foi iniciado um processo de desenvolvimento de um software capaz de processar e otimizar a sua atividade. Neste sentido, a solução idealizada deverá permitir gerir toda a informação sobre as empresas, produtos, compras e envios dos produtos.

Desta aplicação farão parte três módulos com responsabilidades distintas que permitirão aumentar a segurança e produtividade da G4B². Estes módulos devem ser implementados para serem incorporados no software da empresa.


Módulo de transações

Este módulo deve receber a informação de todas as encomendas processadas com sucesso num determinado dia e tem por responsabilidade o registo das transações no livro-razão da empresa (fazendo uso da API do trabalho prático 1). Cada encomenda deverá corresponder a duas transações no livro-razão: uma transação identifica o envio dos produtos enquanto a segunda identifica o pagamento. Neste sentido, como é assumido que as transações recebidas por este módulo foram processadas com sucesso, deve ser adicionado automaticamente um *token* ao responsável (emissor) da transação de modo a poder ser adicionada ao livro-razão.

Este módulo deve ainda permitir a análise do histórico de transações com métricas estatísticas, nomeadamente:

- Número médio de produtos por transação;
- Valor médio das transações;
- Valor médio de vendas e compras por distrito;
- Cada grupo deve acrescentar duas métricas que devem ser do interesse da empresa.

Estas métricas estatísticas devem ser exportadas para um documento em formato JSON.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático II	Ano letivo 2021/2022	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software II	Duração	

Módulo de calculo de custos

Este módulo deve receber uma encomenda e retornar o custo de envio considerando os produtos que a incorporam. Deve ser aplicado um custo de 0.25 euros por quilograma/quilómetros entre distritos considerando a tabela abaixo. A este custo deve ser adicionado um custo de 0.15 euros por m³.

	Viana do Castelo	Braga	Vila Real	Bragança	Porto	Viseu	Aveiro	Guarda	Coimbra	Castelo Branco	Leiria	Lisboa	Santarém	Portalegre	Setúbal	Évora	Beja	Faro
Viana do Castelo	0	60	160	270	75	200	150	270	200	331	250	385	320	365	420	478	520	620
Braga		0	100	217	55	180	125	250	175	309	234	363	300	340	400	450	500	600
Vila Real			0	110	100	100	160	170	180	260	260	400	330	350	430	490	535	630
Bragança				0	210	200	270	180	290	270	350	480	420	360	520	460	540	720
Porto					0	120	75	200	120	260	180	310	250	300	350	400	450	550
Viseu						0	85	75	90	170	160	290	220	260	325	380	427	527
Aveiro							0	160	65	200	115	250	180	230	290	340	390	490
Guarda								0	160	95	255	310	250	180	350	280	365	550
Coimbra									0	140	70	200	140	170	240	300	340	440
Castelo Branco										0	170	224	160	100	260	195	280	460
Leiria											0	140	80	170	180	240	280	380
Lisboa												0	80	220	50	130	180	280
Santarém													0	160	120	160	210	300
Portalegre														0	200	100	180	380
Setúbal															0	190	180	380
Évora																0	80	220
Beja																	0	147
Faro																		0

Tabela 1. Distância entre distritos (Km)

Modulo de expedição

Este módulo recebe as encomendas e atribui-as, agrupadas por distrito, a contentores de camiões com uma capacidade máxima de 63 m³. O resultado deverá ser exportado para um documento JSON com a listagem dos contentores necessários e as encomendas que lhe foram atribuídas.


Resultados

A API produzida para o trabalho prático 1 deve ser corrigida de acordo com os problemas reportados. Para o efeito o código original será disponibilizado no Moodle da U.C.

Os módulos identificados devem ser disponibilizados através de uma API desenvolvida na linguagem Java.

Deve produzir um protótipo que utilize estes módulos para testes e deverá consumir a informação das encomendas importada através de documentos JSON como exemplificado de seguida.

```
{
  "orders": [
    {
      "id": "20210101_01",
      "date": "2021-01-01",
      "sender": {
        "name": "John Doe",
        "address": "street 1",
        "district": "Porto",
        "vat": "12345679"
      },
      "receiver": {
        "name": "Jane Smith",
```

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático II	Ano letivo 2021/2022	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software II	Duração	

```

    "address": "street 2",
    "district": "Lisboa",
    "vat": "234567890"
  },
  "products": [
    {
      "id": 1,
      "name": "product 1",
      "description": "description product 1",
      "price": 10.25,
      "quantity": 10,
      "volume-m3": 2,
      "weight-kg": 5
    },
    {
      "id": 2,
      "name": "product 2",
      "description": "description product 2",
      "price": 7.45,
      "quantity": 15,
      "volume-m3": 0.5,
      "weight-kg": 2.5
    }
  ]
}

```

Tabela 2. Exemplo ficheiro de encomendas


Objetivos

Pretende-se que os estudantes demonstrem as suas competências relativamente ao desenvolvimento ágil de software, com foco em práticas que assegurem a garantia de qualidade de software, nomeadamente: i) **testes de software**; ii) implementação de práticas de *Software Configuration Management*; e; iii) realizar, periodicamente, **inspeção de código**. Pretende-se que os estudantes sigam uma abordagem baseada em *Continuous Integration*, procurando **suportar e automatizar** o mais possível o processo de construção de software.

No final do trabalho, os estudantes deverão ver reforçada a sua capacidade para:

- Analisar um problema e identificar os requisitos, as regras de negócio e cada caso de uso;
- Especificar e executar um conjunto de casos de testes legível;
- Distinguir testes de caixa negra e testes de caixa branca (com a identificação da cobertura) e escolher adequadamente quais os testes que melhor se aplicam a cada caso de uso;
- Usar metodologias de desenvolvimento ágil no desenvolvimento de software e o uso de ferramentas de suporte;
- Estabelecer um plano de desenvolvimento maduro envolvendo uma equipa de trabalho;
- Inspeccionar o código, regularmente, através da identificação, recolher e análise⁽¹⁾ de um conjunto específicos de métricas relacionadas com a qualidade do produto de software, relacionados com outros fatores de qualidade que não o comportamento do sistema.

⁽¹⁾ A tarefa de análise das métricas será elaborada na Unidade Curricular de Matemática Computacional II, através de métodos estatísticos, correlações e regressões lineares, entre outros.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático II	Ano letivo 2021/2022	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software II	Duração	

A qualidade de software, assenta em princípios e práticas de melhoria contínua. Neste contexto, é fundamental, para além da implementação de medidas de correção, procurar a causa dos problemas para que se possam evitar que ocorram novamente no futuro. Uma análise mais extensiva das métricas de qualidade permitirá tomar decisões de ajuste do processo de desenvolvimento de software, contribuindo para a melhoria contínua do processo e produto de software.

Face ao exposto, os estudantes deverão:

1. Planear e acompanhar a execução do projeto utilizando metodologias ágeis, em particular a metodologia SCRUM;
2. Criar e gerir todos os artefactos necessários e associados a uma gestão ágil do projeto (exemplo: *backlogs*, *user stories*, sprints, etc.);
3. Estabelecer mecanismos de suporte às principais tarefas de *Software Configuration Management*. Por exemplo: controlo de versões, controlo de alterações, configuração de mecanismos de *building*, e auditorias;
4. Documentar o processo de análise e especificação de testes;
5. Recolher e analisar um conjunto de métricas relacionadas com a qualidade do produto de software, no âmbito das atividades de inspeção de código.

Para além do exposto nos pontos anteriores (1 a 5), será objeto de avaliação a coerência dos procedimentos e políticas definidas com o suporte tecnológico do processo, bem como o plano SCM subadjacente ao projeto.

Ferramentas

No contexto do desenvolvimento deste projeto o grupo de desenvolvimento terá ao seu dispor a plataforma Gitlab (gitlab.estg.ipp.pt).

O desenvolvimento do trabalho prático compreende também o uso de outras ferramentas como:

- Java – Linguagem de programação usada para o desenvolvimento do motor de pesquisa e especificação de testes;
- Gradle – Ferramenta de configuração e automação de *builds* de software;
- JUnit – Framework para codificação dos testes especificados;
- Jacoco – para análise de cobertura do código, efetuando a contagem da percentagem de cobertura dos testes, face ao total de instruções, ramos, linhas de código, métodos e classes.
- PMD – Ferramenta para análise estática de código e *checkstyle*;
- Gitlab, para (por exemplo) suportar:
 - Controlo de versões do código fonte;
 - Gradle para a configuração de *builds*;
 - Implementação de conceitos de CI durante o desenvolvimento do projeto como *builds* e testes automáticos;
 - *Issue tracker* e utilização *boards* para desenvolvimento usando metodologias ágeis;

O grupo de trabalho deve tirar partido as ferramentas disponibilizadas da melhor forma possível e tendo em consideração os objetivos deste trabalho prático. A figura 1 apresenta uma visão geral de algumas etapas de *Software Quality Assurance* do ciclo de desenvolvimento de software.

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático II	Ano letivo 2021/2022	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software II	Duração	

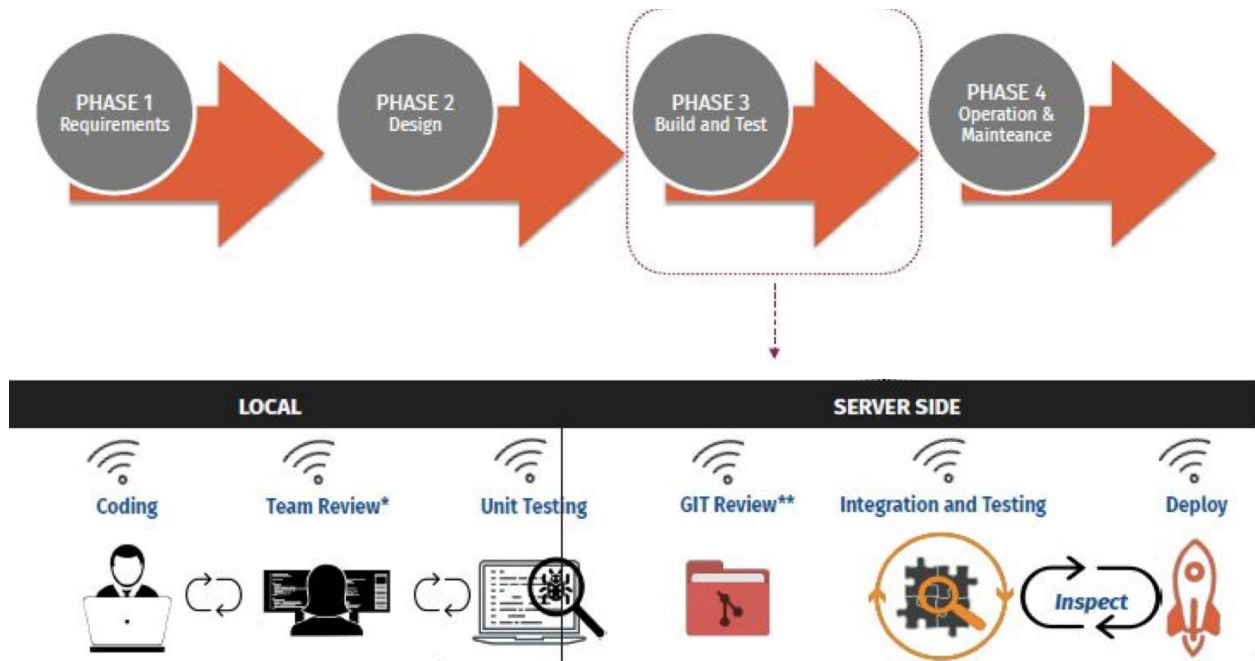


Figura 1. visão geral de algumas etapas de *Software Quality Assurance*

O processo de desenvolvimento será baseado em SCRUM. Os diferentes grupos de trabalho devem estabelecer um *product backlog* organizando num conjunto de *user stories* que serão tratados ao longo do desenvolvimento do projeto. Devem durante a execução do projeto recorrer a gestão de *issues* para gerir o desenvolvimento ágil do projeto em equipa.

Devem ser claramente respeitados os eventos da metodologia, sendo a etapa de revisão e retrospectiva do sprint efetuada durante as aulas práticas na presença de um docente da unidade curricular.


Devem também organizar as tarefas de cada sprint usando *Scrum Boards* e acompanhar a evolução da implementação de cada implementação e assegurar a qualidade do código desenvolvido.

A cada sprint semanal, deverá existir software funcional no *master branch*, por bem simples que seja.

Semanalmente, aquando das etapas para a *release* semanal, as seguintes tarefas deverão ser despoletadas automaticamente:

- Execução dos testes de software definidos e geração do respetivo relatório de testes;
- Análise de cobertura e geração do respetivo relatório.
- Inspeção de código, focada essencialmente em aspetos de *design* e *documentation*, verificando: i) cyclomatic complexity; ii) cognitive complexity; iii) NPath Complexity; iv) padrões de documentação; v) duplicação de código; vi) nº de linhas por método; vii) n.º de linhas por método; viii) Grau de *coupling*.

A informação que consta dos relatórios de teste, análise de cobertura e inspeção de código, deverá ser, semanalmente, compilada num único documento, de acordo com a estrutura a definir na UC de MCII e entregue ao responsável dessa mesma UC. Posteriormente, esses dados farão parte de um *dataset* mais robusto e que será alvo de análise estatística.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático II	Ano letivo 2021/2022	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software II	Duração	

Apresentação do trabalho

A apresentação deverá resumir o relatório e apresentar essa informação num formato que permita ao avaliador perceber o processo e os artefactos construídos.

A apresentação do trabalho deve ser feita com recurso a um vídeo (2 min máximo) demonstrando o modelo de desenvolvimento, passando por todas as fases necessárias e utilizando todas as ferramentas envolvidas.

Submissão do Projeto

Todos os artefactos gerados no decorrer deste projeto devem ser alvo de submissão via plataforma moodle num ficheiro zip com a seguinte denominação ESII-GrupoXX-TP2.zip onde os caracteres XX devem ser substituídos pelo número atribuído ao grupo. Devem indicar (nos recursos submetidos) o endereço Gitlab do projeto e devem dar permissões de visualização aos docentes: CDS, MFG, JMRC e OAO.

CrITÉRIOS de Avaliação

Sendo um trabalho para alunos inscritos em avaliação contínua, a avaliação do trabalho prático será desenvolvida semanalmente, nas aulas ou através da plataforma de gestão do projeto, o GitLab.

Os artefactos em avaliação ao fim de cada sprint são:

- *User Stories* implementadas
- Especificação das *user stories*
- Código fonte da aplicação (em JAVA)
- Código fonte dos testes (em junit 5)
- Análise dos testes efetuados
- Gestão e uso do repositório Git
- Cumprimento das normas identificadas no SCM
- Resultado das atividades de inspeções de código


A juntar aos parâmetros avaliados ao fim de cada sprint existirá uma avaliação no final do projeto dos artefactos finais produzidos.

Grupo de trabalho

A constituição dos grupos de trabalho será igual aos grupos já definidos no primeiro trabalho prático da unidade curricular desde que não exceda os **3 elementos**.

Cronograma de atividades

Considerando que deverão seguir uma abordagem ágil ao desenvolvimento do projeto, a equipa docente parte do princípio de que são capazes de incorporar os princípios e valores ágeis ao desenvolvimento do projeto. Assim, enquanto "*self-organized teams*", deverão ser capazes de delinear as vossas próprias *baselines*. Caberá à equipa docente avaliar a exequibilidade e comprimento das *baselines*, de acordo com o SCM Plan. Contudo, importa referir que algumas das boas práticas SCM devem ser cumpridas, nomeadamente: i) escolher e aplicar um padrão SCM para gestão das *codelines*, sendo que, o desenvolvimento deverá ser feito com poucas *codelines*; iii) Testar e integrar recorrentemente e o mais cedo possível, isto implica que devem existir *commits* regulares (definir período); iv) automatizar, e; v) clarificar papeis e responsabilidade. A estratégia SCM deverá estar alinhada com os princípios ágeis de desenvolvimento de software.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático II	Ano letivo 2021/2022	Data
	Curso Licenciatura em Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software II	Duração	

Datas importantes

- Fim do projeto: 21 de Janeiro
- Discussões: 22 a 29 de Janeiro