

Relatório do segundo seminário de ES2

Alunos:

Daniel Marcondes Bougleux Sodré

José Victor de Paiva e Silva

André Fernandes Gonçalves

Juan Müller Pereira Evangelista

Lucas Tavares Sousa

1. Detalhamento das sprints

1.1 - 2ª Sprint:

1.1.1 - Lista de Atividades:

O principal objetivo desta sprint foi a inicialização do código, estruturar e gerar as funcionalidades do mapa da partida, tratar as ações que um jogador pode realizar em seu turno, tratar das telas não relacionadas à partida e suas funcionalidades respectivas. Estas atividades são: Componentes do Mapa, Componentes do Menu, Organização e Funcionalidades do Mapa, Histórico de Partida, Combate, Login, Cadastro, Recebimento de Tropas.

1.1.2 - Atividades Cumpridas:

As atividades concluídas levaram mais tempo do que o esperado, e por conta disso essa sprint acabou ficando acima do orçamento. As atividades concluídas foram: Componentes do Mapa, Componentes do Menu, Organização e Funcionalidades do Mapa, Histórico de Partida, Login, Chat e Criação de partida.

1.1.3 - Atividades Pendentes:

Não conseguimos trabalhar no projeto todos os dias planejados, e por conta disso algumas atividades ficaram pendentes. Elas também fazem parte do caminho crítico, e por conta disso o projeto como um todo está atrasado. São as atividades: Combate e Recebimento de Tropas.

1.2 - 3ª Sprint:

1.2.1 - Lista de Atividades:

O principal objetivo desta sprint é a completude das funcionalidades da partida e do saguão, assim como a implementação de parte da Inteligência Artificial dos Bots. As atividades são: Movimentação de Tropas, Troca de Cartas, Criar Partida, Chat e Perfil do Jogador,

Estratégia de Destruir o exército de uma cor específica, Estratégia de conquistar 18 territórios com no mínimo 2 tropas em cada, Estratégia de conquistar os continentes Europa + América do Sul e Oceania + 1 Continente Qualquer, Estratégia de conquistar América do Norte + África ou Oceania.

1.2.2 - Atividades Cumpridas(11/08):

Como essa sprint ainda não terminou, está sendo considerado apenas as atividades que foram planejadas para serem feitas até o dia 11 de Agosto. Tais atividades concluídas são: Chat e Criar Partida.

1.2.3 - Atividades Pendentes(11/08):

Não conseguimos trabalhar no projeto todos os dias planejados, e por conta disso algumas atividades ficaram pendentes. São as atividades: Perfil do Jogador, Movimentação de Tropas, Troca de Cartas. A atividade Troca de Cartas faz parte do caminho crítico, e por conta disso o projeto como um todo está atrasado

1.2.4 - Atividades a serem feitas após 11/08:

Como essa sprint ainda não terminou, ainda existem atividades para serem feitas, porém que não são contadas como pendentes pois sua data de entrega ainda não chegou. São as atividades: Estratégia de Destruir o exército de uma cor específica, Estratégia de conquistar 18 territórios com no mínimo 2 tropas em cada, Estratégia de conquistar os continentes Europa + América do Sul e Oceania + 1 Continente Qualquer, Estratégia de conquistar América do Norte + África ou Oceania.

2. Burndown das sprints

O gráfico de burndown, é um meio que temos como avaliar um andamento de um sprint. Para isso, o gráfico tem duas linhas, a primeira (no nosso caso em azul), mostra uma diminuição linear quantidade de tempo restante em horas planejadas para a sprint, a segunda linha demonstra a quantidade de horas restante reais, ou seja, a soma das horas planejadas das atividades não terminadas.

Para fazer esse gráfico foi preciso calcular, primeiramente, a quantidade total de horas e de dias da sprint para fazer a redução linear e, além disso, conforme terminavam as atividades colocamos essa horas em uma planilha para no término da sprint montarmos o gráfico da quantidade restante de horas reais.

2.1 - Burndown da Segunda Sprint:

A Figura 1 mostra o burndown da segunda sprint. Durante o período da segunda sprint, não conseguimos resolver as atividades planejadas, que acabaram sendo resolvidas no tempo da terceira sprint.

Gráfico BurnDown: Segunda Sprint

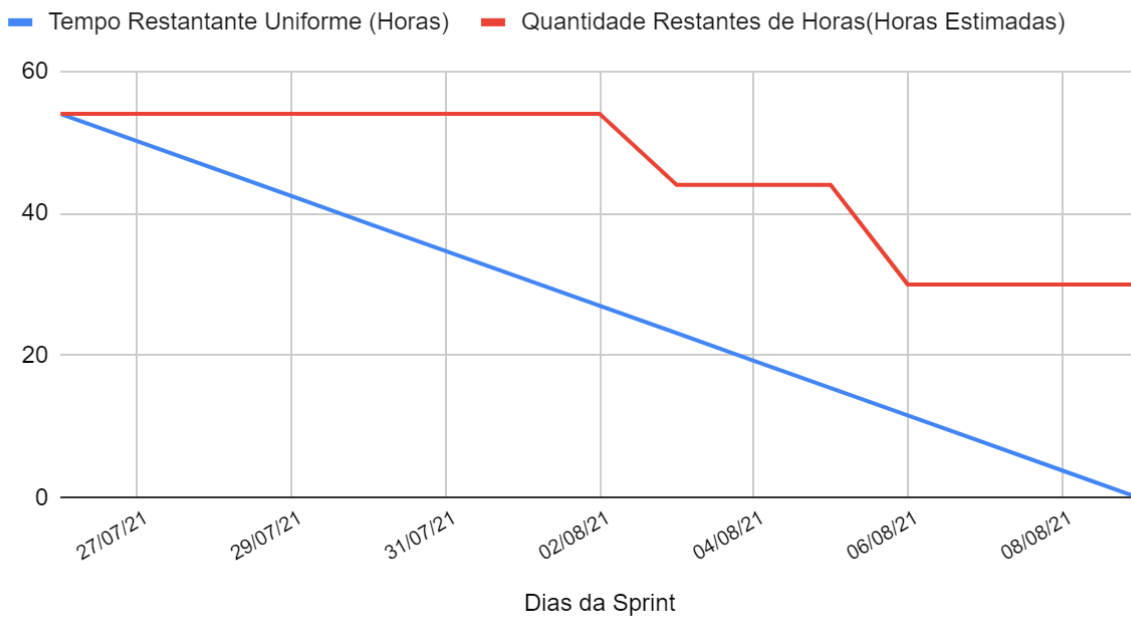


Figura 1: Gráfico burndown da segunda sprint.

2.2 - Burndown da Terceira Sprint:

Já na Figura 2, que detalha a terceira sprint, podemos ver que no momento atual, a sprint ainda está em andamento. Ela acabou herdando atividades incompletas da segunda sprint, e ainda assim, há atividades pendentes tanto da segunda sprint quanto da terceira.

Gráfico BurnDown: Terceira Sprint

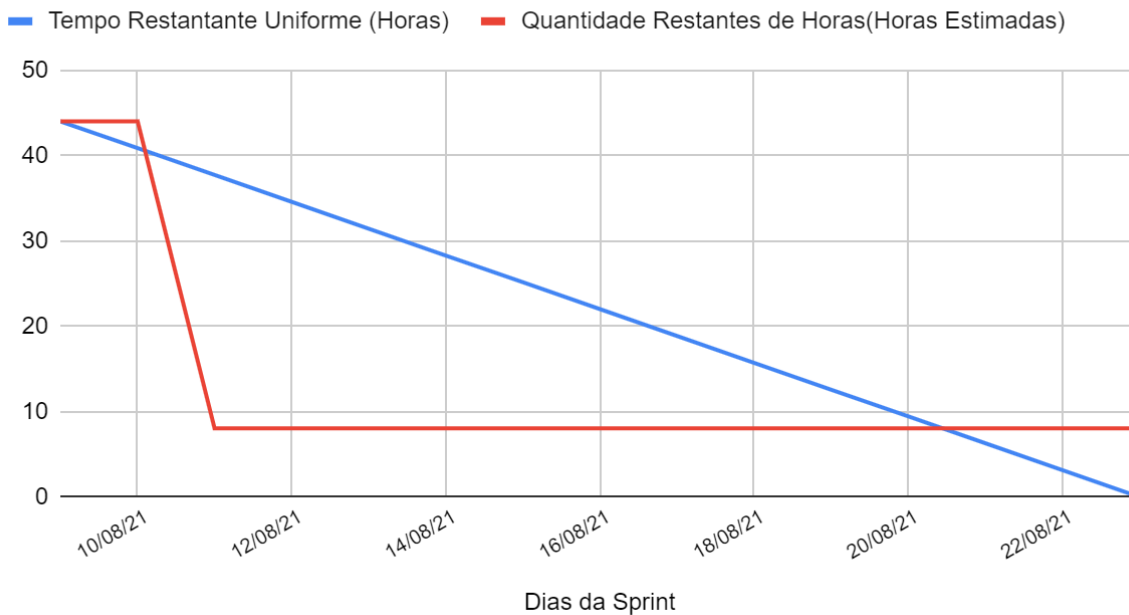


Figura 2: Gráfico burndown da terceira sprint.

Segue o link da planilha: [BurnDown](#)

3. Análise de Valor Agregado

O gráfico de Valor Agregado, possui como principal objetivo avaliar o projeto em questão nos gastos do projeto. Em primeiro lugar calculamos a porcentagem planejada de conclusão do projeto com o passar do tempo e por sua vez a utilizamos para calcular os gastos planejados do projeto em função do tempo, esse dado é chamado de Preço Planejado(PV). Quando o projeto começa a ser desenvolvido, ao terminar uma tarefa adicionamos a porcentagem de conclusão do projeto referente aquela tarefa em uma planilha com o dia do seu término, e da mesma forma calculamos o gasto dessas tarefas por dia, esse valor é chamado de Valor Agregado(EV). Por fim, repetimos o processo para calcular o Valor Agregado(EV), mas utilizando o tempo realmente gasto no projeto, dessa forma calculamos o Preço Real(AC).

Gráfico de Valor Agregado

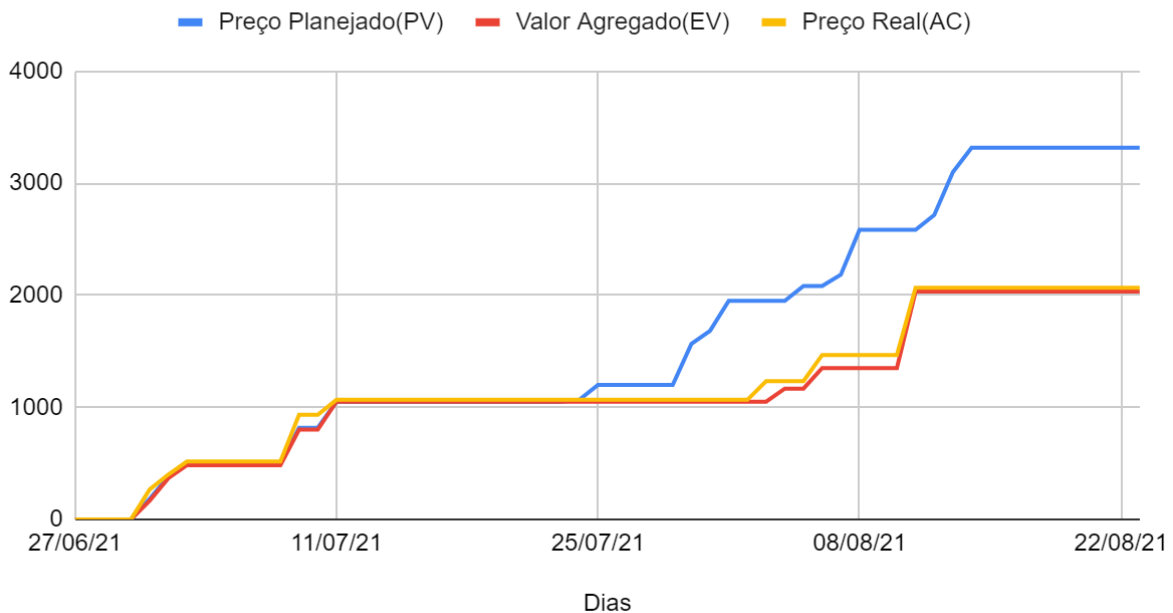


Figura 3: Gráfico de valor agregado do projeto.

Valores Atuais de:

PV: R\$ 2.583,85

EV: R\$ 2.033,74

AC: R\$ 2.067,08

Conclusões matemáticas:

$$SPI = \frac{ev}{pv} = \frac{2033,74}{2583,85} = 0,78 < 1, \text{ portanto, estamos atrasados.}$$

$$CPI = \frac{ev}{ac} = \frac{2033,74}{2067,08} = 0,98 < 1, \text{ portanto, estamos gastando ligeiramente mais do que foi planejado.}$$

Com o SPI e CPI calculados é claro que estamos bastante atrasados e gastando ligeiramente mais que o planejado. Isso vem da dificuldade do grupo em conciliar os horários para trabalhar em conjunto e atribuir as atividades devidas a cada integrante.

Segue o link da planilha: [Valor Agregado](#)

4. Controle de modificações

Para controlar as modificações, já que os requisitos foram elucidados logo no começo, e não existe um cliente real, não foi escolhida nenhuma ferramenta para auxiliar no controle de modificações. As modificações no projeto foram apenas a exclusão do recurso Configurações do jogo e a adição do cadastro de usuário. A

exclusão do recurso Configuração ocorreu pois não era pertinente alterar volume do jogo ou modo de tela, se comparado ao esforço e custos necessários, enquanto a inclusão do Cadastro facilitou a inclusão de usuários no sistema e não demandou mais esforço ou custos.

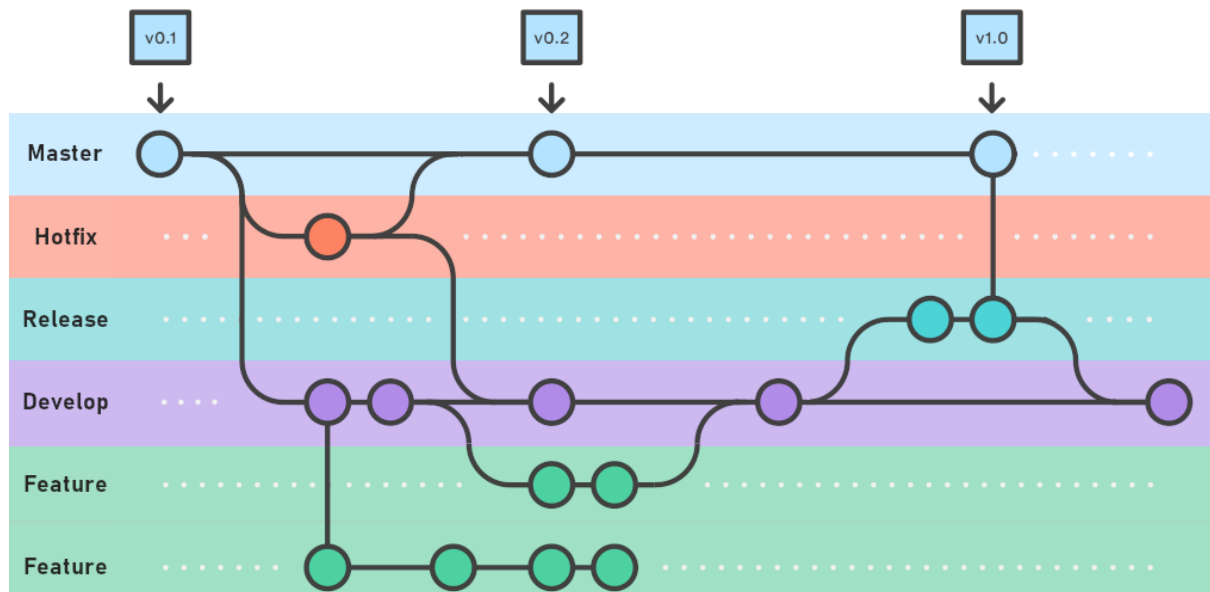
Para tal, foi feita uma análise em equipe para ver se tais mudanças prejudicariam o andamento do projeto e seus custos, onde concluímos que não atrasariam o projeto nem o tornaria mais caro. Já que as modificações ocorreram em estado inicial de codificação, não houve necessidade de alterar trechos de código, apenas documentação.

5. Estratégia de ramificação

Nossa estratégia de ramificações faz parte do fluxo de trabalho gitflow. Nesta estratégia há duas ramificações de destaque: a master e a development. Na master fica a versão do produto pronta para uma entrega, enquanto a development é onde os desenvolvedores incluem suas soluções para cada recurso. Quando a development adquire recursos suficientes, ou aproxima-se de uma data de lançamento, a development faz merge com a master.

Além disso, existem 4 tipos de branches auxiliares, que são feature, release, bugfix e hotfix que tem as seguintes funções.

- feature: Branch responsável para realizar o desenvolvimento de uma certa funcionalidade.
- release: Branch responsável a fazer uma ponte entre a master e a develop, nessa branch é que normalmente realizam os testes antes do produto ir para produção.
- Bugfix: Branch responsável por realizar correção de bugs/erros encontrados na release, geralmente erros encontrados nos testes.
- Hotfix: Branch responsável por realizar correções de bugs/erros encontrados em produção, geralmente reportado pelo cliente.



6. Contribuições no repositório

Todos os integrantes participaram ativamente do desenvolvimento do projeto. Para o desenvolvimento, foi utilizado o recurso Live Share da IDE Visual Studio Code para a codificação simultânea, e o recurso de compartilhamento de telas do aplicativo Discord. Assim, fomos capazes de codificar as soluções juntos e compartilhar as ideias de como resolver os problemas..

O versionamento do código, porém, acontece apenas em uma máquina, sendo assim apenas um dos desenvolvedores pode *commitar* a solução. Por convenção do grupo, decidimos que os integrantes André, Daniel e Lucas seriam os principais responsáveis por atualizar as soluções no repositório.

7. Conteúdo do repositório

Repositório no GitHub: <https://github.com/FBMLJ/es2-trabalho-war>

Versão Parcial do Produto: A demonstração do produto que temos até agora está no vídeo entregue junto com o relatório.