

5 MATERIAIS E MÉTODOS

Nessa seção são descritos e discutidos os materiais e métodos utilizados nesse trabalho. A primeira subseção tem como objetivo descrever a metodologia para dar suporte no processo do jogo. A segunda subseção apresenta a arquitetura *Model View Controller* (MVC), base para o desenvolvimento do jogo Nonda. A partir da terceira subseção são descritos em detalhes, ao longo de 15 subseções, todas características do jogo que ajuda entender no ambiente de games o processo de software *Scrum Solo*.

5.1 Processo de Software *Scrum Solo*

Esse projeto foi desenvolvido de acordo com o *framework Scrum solo*, que tem base no *Scrum* e é uma metodologia Ágil. Dentre as fases do processo estão: detalhes dos requisitos, análise & design, implementação & teste, aprovação do teste, reavaliar / priorizar novas tarefas. Segundo JAMES, M. (2010), no desenvolvimento de jogos, o *Scrum* tem início a partir do levantamento dos requisitos passando pelo análise e design do UI/UX, ambiente, personagens e animação passando pela implementação da lógica do jogo, até a fase de teste onde finaliza a primeira iteração (*sprint*), sendo que nessa iteração o foco principal é estabelecer quais métodos, instâncias serão reaproveitados para reutilizar nas demais fases. O *Scrum solo*, desenvolvido por professores da UTFPR-CP, beneficia o desenvolvedor solo e tem as boas práticas do *Scrum* e *Personal Software Process* (PSP), processo que auxilia o desenvolvedor a entender a própria performance, desenvolvendo melhorias em suas práticas de trabalho. O *Scrum solo* divide o projeto em várias sprints (iterações nos ciclos de desenvolvimento) que duram 1 semana e tem incrementos cada vez que um product owner identifica uma nova prioridade no product backlog (funcionalidades que o scrum team deve desenvolver no software). Durante as iterações, o desenvolvedor codifica e testa pequenas partes do projeto. O diferencial é que entrega uma fatia do software no prazo dito acima, e se necessário existe reunião para orientação entre o grupo de validação (cliente final) e o desenvolvedor e não há reuniões diárias, como acontece no *Scrum* (FABRI, J. A. et al, 2016, no prelo).