

Contextualização dos processos regulatórios na aprendizagem de programação

Fases, Áreas e Processos para Aprendizagem Regulada/Contextualização em Programação					
	Tipo de Regulação	Áreas para Regulação			
Fases		Cognitivo/ Sócio-cognitivo	Motivação/Afeto	Comportamento	Contexto
Fase 1. Previsão, planejamento e ativação	Autorregulação	Definindo meta	Predispondo-se para alcançar o objetivo	Planejando tempo e esforço	Percebendo a tarefa
		Avaliando conhecimento prévio sobre conteúdo	Julgando autoeficácia	Planejando o auto-monitoramento do comportamento.	Percebendo o contexto
Planejamento antes da Codificação	Corregulação	Ativando metacognição	Percebendo a dificuldade da tarefa	Usando Kanban para planejamento de tarefas de programação.	Estabelecendo estratégias para executar e monitorar tarefas de programação.
		Detectando itens de problemas importantes antes da codificação.	Percebendo o valor da tarefa	Utilizando Kanban para autoavaliação de conteúdos teóricos e práticos de programação.	
Fase 2. Monitoramento e avaliação	Autorregulação	Desenhando a solução antes de codificar.	Ativando o interesse		
		Usando diagramas para explicar o design da solução e suas conexões com o código.	Vendo Obstáculos de uma Perspectiva Positiva na Programação		
Monitoramento durante Codificação e Teste	Corregulação	Praticando e entendendo os fundamentos antes de começar a programar.			
		Estabelecendo entendimentos compartilhados de demandas de tarefas, negociando o significado do problema e estabelecendo metas.	Antecipando boas relações no grupo.	Criando fluxos de trabalho para atingir metas, incluindo a definição de cronogramas.	Negociando e descrevendo papéis de acordo com o perfil do aluno.
Fase 3. Avaliação e feedback	Autorregulação	Interagindo com os membros da equipe sobre as ações a serem executadas.	Encorajando futuras participações e interações.	Negociando a divisão do trabalho	Organizando a equipe (protocolo de comunicação/regras de engajamento).
		Estabelecendo uma compreensão compartilhada dos conceitos de programação.	Usar frases motivadoras de bom humor na programação.	Usando Scrum para planejar tarefas de programação colaborativa.	Escolhendo tecnologias de groupware para programação.
Fase 4. Aprendizagem e reflexão	Corregulação				Planejando uma escrita de programação colaborativa.

	Corregulação	<p>Monitorando o entendimento compartilhado.</p> <p>Acompanhando os processos gerais do grupo.</p> <p>Acompanhando do avanço do conhecimento.</p> <p>Detectando erros e verificando a plausibilidade.</p> <p>Detectando conflitos sócio-cognitivos no grupo.</p> <p>Entendendo padrões de programação juntos.</p> <p>Experienciando padrões de programação juntos.</p> <p>Monitorando a resolução de problemas colaborativa em programação.</p>	<p>Monitorando a motivação do grupo para a participação e interações.</p> <p>Detectando conflitos sócio-emocionais em grupo.</p> <p>Acompanhando do comprometimento do grupo na programação</p>	<p>Acompanhando objetivos e progresso do grupo.</p> <p>Usando fluxos de trabalho para monitorar o progresso das atividades.</p> <p>Usando Scrum para monitorar tarefas de programação colaborativa.</p>	<p>Monitorando a mudança de funções e protocolos de comunicação.</p> <p>Acompanhando regras de engajamento.</p> <p>Monitorando o contexto de programação colaborativa.</p>
Fase 3. Controle Codificação e Teste	Autorregulação	<p>Selecionando e adaptando estratégias cognitivas para aprender, pensar</p> <p>Adaptando padrões de programação.</p> <p>Combinando padrões de programação.</p>	<p>Selecionando e adaptando estratégias para gerenciar a motivação e o afeto.</p> <p>Reduzindo a Ansiedade em programação.</p>	<p>Aumentado/diminuindo o esforço</p> <p>Persistindo/desistindo</p> <p>Comportamento de busca de ajuda</p> <p>Usando Kanban para gerenciamento de tarefas em programação</p>	<p>Alterando ou renegociando tarefas</p> <p>Alterando ou abandonando o contexto</p> <p>Atuando no contexto individual de programação.</p>
	Corregulação	<p>Comunicando-se com os membros da equipe sobre as ações que estão sendo executadas.</p> <p>Fazendo planos colaborativos para atingir metas, incluindo a seleção de estratégias sócio-cognitivas.</p> <p>Descobrindo o tipo de colaboração, interação para resolver o problema, juntamente com os objetivos.</p> <p>Avançando e explicando soluções. Coordenação de conflitos sócio-cognitivos.</p> <p>Controlando o progresso geral das soluções do grupo.</p>	<p>Controlando a quantidade e a qualidade da participação e das interações do grupo.</p> <p>Fornecendo feedback sobre as participações e interações do grupo.</p> <p>Evitando e controlando os conflitos sócio-emocionais no grupo.</p> <p>Promovendo o respeito ao criticar o ponto de vista do outro.</p> <p>Promovendo a participação na programação.</p> <p>Desenvolvendo relações de confiança na programação.</p>	<p>Procurando ajuda do professor quando um conflito de ideias não chega a um consenso</p> <p>Gerenciando workflows.</p> <p>Usando Coding DOJO (Kata) em programação introdutória.</p> <p>Usando Coding DOJO (Randori) em programação introdutória.</p>	<p>Controlando os papéis no grupo e protocolos de comunicação</p> <p>Fornecendo feedback sobre os papéis no grupo e protocolos de comunicação.</p> <p>Analizando prós e contras em programação.</p> <p>Atuando no contexto da programação colaborativa.</p>

		<p>Facilitando a crítica e a construção das perspectivas dos outros.</p> <p>Subdividindo o problema computacional.</p> <p>Analisando e construindo artefatos de software de terceiros.</p> <p>Codificando juntos a partir de experiências anteriores.</p>			
<p>Fase 4. Reação e Reflexão</p> <p>Reflexões sobre a Codificação do Programa</p>	Autorregulação	<p>Julgamentos cognitivos</p> <p>Pensamento crítico e metacognição</p> <p>Aprendendo com os erros e acertos na programação.</p>	<p>Reações afetivas</p> <p>Objetivos intrínsecos e extrínsecos, valor da tarefa, crenças de controle, autoeficácia e teste de ansiedade.</p> <p>Refletindo sobre a motivação do aluno na programação.</p>	<p>Comportamento de escolha</p> <p>Regulação do esforço</p> <p>Busca de ajuda</p> <p>Tempo/ambiente de estudo</p> <p>Usando Kanban para refletir sobre tarefas de programação.</p> <p>Refletindo sobre os prós e contras do Kanban para o progresso individual na programação.</p>	<p>Avaliação das tarefas</p> <p>Avaliação do contexto</p> <p>Aprendizado entre pares, tempo/ambiente de estudo</p> <p>Refletindo sobre prós e contras na programação.</p> <p>Refletindo sobre o contexto de programação individual.</p>
	Corregulação	<p>Refletindo e reparando o entendimento compartilhado.</p> <p>Avaliando soluções conjuntas atuais.</p> <p>Refletindo sobre diferentes pontos de vista.</p> <p>Acompanhando os resultados das ações e avaliação do sucesso na resolução do problema.</p> <p>Refletindo sobre os objetivos do grupo, progresso e conquistas.</p> <p>Fazendo adaptações a metas, planos ou estratégias colaborativas.</p> <p>Refletindo sobre diferentes soluções de computação.</p>	<p>Avaliando os aspectos emocionais dos membros do grupo, no que diz respeito ao respeito mútuo e ao engajamento nas atividades do grupo.</p> <p>Avaliação do grupo quanto ao número de interações e quantas pessoas diferentes interagiram.</p> <p>Prevenindo a falta de participação e interações.</p> <p>Refletindo sobre a motivação do grupo na programação.</p> <p>Refletindo sobre as relações de confiança na programação.</p>	<p>Refletindo sobre os objetivos e o progresso do grupo.</p> <p>Refletindo sobre fluxos de trabalho para verificar a produtividade.</p> <p>Adaptando os fluxos de trabalho.</p> <p>Refletindo sobre os prós e contras do Scrum para a programação colaborativa.</p>	<p>Refletindo sobre os papéis do grupo e protocolos de comunicação.</p> <p>Adaptando funções de grupo e protocolos de comunicação.</p> <p>Refletindo sobre o contexto de programação colaborativa.</p>