

**Diretriz Curricular de Desenvolvimento Java (ênfase en React)**

|  |
| --- |
| **Objetivos de Desempenho** |
| **Formandos em Desenvolvimento Java serão capazes de:** |
| a. Resolver problemas de forma eficaz para identificar soluções para resolver questões relativas ao projeto. |
| b. Usar as melhores práticas de codificação para executar as tarefas atribuídas |
| c. Realizar testes de forma consistente e proativa para verificar se o código está correto |
| d. Aprender novas habilidades técnicas e não-técnicas de forma proativa para melhorar seu desempenho e aumentar a probabilidade de uma promoção |
| e. Se comunicar e colaborar de forma eficaz com colegas de trabalho e outras partes interessadas usando uma variedade de plataformas. |

|  |  |
| --- | --- |
| **Momentos Críticos** | |
| **Momento Crítico** | **O que profissionais de alto desempenho fazem para enfrentar esse desafio?** |
| 1) Analisam criticamente uma tarefa e determinam de forma precisa os passos e o tempo para concluí-la | * Fazem perguntas para esclarecer o resumo da tarefa que lhes é apresentada para garantir que a entendem corretamente. * Discutem quanto tempo a tarefa deve durar e alinham datas de conclusão. * Encontram a melhor abordagem para resolver o problema * Compartilham resultados iniciais e em curso com sua equipe e supervisor para garantir que todos estão alinhados em relação à forma como a tarefa deve ser concluída. |
| 2) Determinam a melhor abordagem para resolver um problema (p.ex. resolvem o problema de forma independente, pedem o suporte de colegas e supervisores, etc.) | * Percebem quanto tempo devem dedicar para tentar resolver um problema técnico de forma independente antes de pedirem ajuda * Julgam quando pedir o suporte de outros, e quem é a melhor pessoa para pedir * Potencializam e priorizam diferentes fontes de conhecimento para encontrar a melhor solução para seus problemas (p.ex. usar fóruns online, colegas de trabalho, recursos criados pela empresa, etc.). |
| 3) Escrevem códigos de forma eficiente e simplificada | * Sabem como codificar mais coisas por conta própria sem ter que pesquisar soluções * Lembram de situações nas quais tiveram que usar uma estrutura de codificação semelhante e as reciclam * Reciclam estruturas de codificação disponíveis online * Organizam códigos em arquivos e pastas para que seja mais fácil encontrá-los e reciclá-los * Usam uma estrutura consistente em todo o código que os outros possa entender facilmente (p.ex. aplicação de funções que diminuem o nível de complexidade) |
| 4) Realizam testes minuciosos para identificar e corrigir erros e prevenir possíveis defeitos | * Pensam nas diversas maneiras em que um erro pode ocorrer e realizam testes para verificar se o código está "blindado" para todas as situações/variáveis/em diversas plataformas e navegadores, sendo meticulosos e prestando atenção aos detalhes * Usam uma variedade de abordagens para evitar erros como pedir a colegas para analisarem seu trabalho, realizar testes unitários, usar uma abordagem DevOps em seu trabalho. |
| 5) Identificam oportunidades para realizar treinamentos adicionais e aprender uma variedade de novas habilidades (p.ex. técnicas de gestão, conhecimento de negócios, tendências de mercado, etc.) simultaneamente enquanto realizam suas principais funções. | * Acompanham o ritmo da curva de aprendizagem esperada * Criam um plano sobre quais novas habilidades são importantes aprender para solucionar problemas mais complexos e avançar em suas carreiras (p.ex. língua inglesa) * Desenvolvem um conhecimento sólido sobre a indústria na qual estão trabalhando, e sabem que problemas de negócios sua empresa está tentando resolver. * Identificam áreas na empresa nas quais possam crescer em sua carreira * Se oferecem para realizar tarefas mais complexas fora do escopo de sua função * Tomam notas e praticam o que aprendem e não fazem as mesmas perguntas diversas vezes * Não desistem quando estão aprendendo coisas complexas * Encontram oportunidades para aprender novas habilidades fora do horário de trabalho (p.ex. em dias de pouco movimento, antes ou depois do trabalho, durante intervalos) |
| 6) Adaptam o estilo de comunicação e escolha de palavras para explicar de forma clara informações técnicas a vários públicos | * Traduzem de forma clara e fluente informações técnicas para conhecimento de negócios para públicos sem conhecimento técnico (p.ex. unidades de negócios). |
| 7) Contribuem com a equipe/empresa/cliente comunicando de forma proativa planos e prazos de projetos, participando em reuniões, sugerindo soluções e oferecendo novo conhecimento e suporte àqueles que precisam. | * São reconhecidos/as como líderes e procurados/as para aconselhar partes interessadas * São bons ouvintes e se destacam ao dar e receber feedbacks * Ensinam os outros sobre novas tecnologias ou outras habilidades que adquiram * Garantem, de forma proativa, o compartilhamento de aprendizagens, melhores práticas e soluções com partes interessadas |

|  |
| --- |
| **Sessões de Habilidades Comportamentais e Mentalidades** |

* Introdução a Habilidades Comportamentais e Mentalidades
* Persistência
* Mentalidade de Crescimento
* Orientação ao Futuro
* Responsabilidade Pessoal
* Trabalho em Equipe
* Comunicação
* Comunicação Não Violenta
* Atenção para Detalhes
* Proatividade

|  |
| --- |
| **Sessões de Aplicação de Habilidades** |

* Estudos de caso
* Dramatização
* Atividade de Treinamento para Manter-se Atualizado

|  |
| --- |
| **Sessões de Habilidades Essenciais para o Emprego** |

* Preparação para o Formulário de Candidatura a Emprego 1
* Construindo uma Presença Digital
* Introdução ao pitch pessoal
* Oficina de pitch pessoal
* Trajetórias 1
* Trajetórias 2
* Trajetórias 3
* Currículo 1: Introdução
* Currículo 2: Oficina
* Introdução a Entrevistas
* Oficina de Entrevistas

|  |
| --- |
| **Sessões de Reforço e Reflexão** |

* Introdução à Avaliação
* Introdução à Tabela de Desempenho e à Análise da Tabela de Desempenho
* Protocolo de Análise da Tabela de Desempenho Semanal
* Introdução à Reflexão Diária
* Protocolo de Reflexão Diária
* Introdução ao Treinamento entre Pares
* Treinamento entre Pares
* Projeto Final

|  |
| --- |
| **Sessões de Apoio e Orientação Social** |

* Planejamento para Cuidado Infantil
* Equilíbrio entre Vida Pessoal e Profissional
* Se Adaptando a Mudanças
* Controle de Estresse
* Mulheres na Tecnologia
* Gestão Financeira 1
* Introdução aos Pontos Fortes de Caráter VIA
* Descoberta dos Pontos Fortes de Caráter VIA

|  |
| --- |
| **Sessões da Comunidade** |

* Protocolo de Abertura Diária
* Introdução à Abertura Diária
* Eleição de Representante
* Agora e Depois
* Mostrar e Explicar
* Mantendo Contato
* Dar e Receber
* Bem-vindos à Generation
* Apresente um Amigo
* Pontos Altos da Vida
* Adivinhe o Fato
* Isolado em uma Ilha
* Notas de Valorização de Colegas
* Etapas do desenvolvimento do grupo

|  |
| --- |
| **Sessões Técnicas** |

|  |  |  |
| --- | --- | --- |
| **#** | **MÓDULO** | **Título da sessão** |
| 0 | Introdução à indústria e cargo | **0.1 - Introdução à indústria e cargo** |
| 0.1.1 - Introdução ao Currículo de Desenvolvimento Java |
| 0.1.2 - Introdução à Função de Desenvolvedor/a Java |
| 0.1.3 - Colaboração e Comunicação Eficaz no Local de Trabalho |
| 0.1.4 - Manter-se Atualizado |
| 0.1.5 - Resolução de Problemas |
| 1 | Introdução à programação | **1.1 - Fundamentos de programação** |
| 1.1.1 Introdução à Ciência da Computação |
| - 1.1.2 - ‘ |
|  |
|  |
| **1.2 - Terminal + Git e GitHub** |
| - 1.2.1 - Versionamento, Markdown e Terminal |
| - 1.2.2 - Git, GitHub e GitFlow |
| - 1.2.3 - Revisão |
|  |
| **1.3 - Agile - SCRUM** |
| - 1.3.1 - Desenvolvimento Ágil de Software |
| - 1.3.2 - Revisão |
|  |
| 2 | HTML, CSS e JS | **- 2.1 - HTML** |
| - 2.1.1- Introdução ao Desenvolvimento Web |
| - 2.1.2 - HTML5 |
| - 2.1.3 - CSS3 |
| - 2.1.4 - Bootstrap |
| - 2.1.5 - Revisão |
|  |
| **- 2.2 - JavaScript** |
| - 2.2.1 - Lógica de Programação com JavaScript |
| - 2.2.2 - Funções |
| - 2.2.3 - Erros e depuração |
| - 2.2.4 - Objetos em JavaScript |
| - 2.2.5 - DOM |
| - 2.2.6 - Multimídia |
| - 2.2.7 - JavaScript Web APIs |
| - 2.2.8 - Revisão |
|  |
| 3 | Web Avançada | **- 3.1 - Node.js** |
| - 3.1.1 - Node.js e NPM |
| - 3.1.2 - Revisão |
|  |
| **- 3.2 - TypeScript** |
| - 3.2.1 - Lógica de Programação com TypeScript |
| - 3.2.2 - Programação Orientada a Objetos com TypeScript |
| - 3.2.3 - Revisão |
|  |
| **- 3.3 - React** |
| - 3.3.1 - Primeiros Passos com o React |
| - 3.3.2 - Rotas e Componentes |
| - 3.3.3 - Hooks, Ciclo de Vida e Axios |
| - 3.3.4 - Ajustes e Deploy do Projeto |
| - 3.3.5 - Revisão |
|  |
| 4 | Java | **- 4.1 - Java** |
| - 4.1.1 - Introdução e Instalação Java |
| - 4.1.2 - Java IDE - IntelliJ Idea |
| - 4.1.3 - Integração de idéias Git + IntelliJ |
| - 4.1.4 - Sintaxe básica |
| - 4.1.5 - Classes e objetos |
| - 4.1.6 - Composição e herança |
| - 4.1.7 - Embalagem |
| - 4.1.8 - Interfaces e classes abstratas |
| - 4.1.9 - Coleções |
| - 4.1.10 - Asserções, anotações, expressões lambda |
| - 4.1.11 - Fluxos de E / S |
| - 4.1.12 - Revisão |
| 5 | SQL e Spring Framework | **- 5.1 - SQL** |
| - 5.1.1 - Banco de Dados Relacionais e MySQL |
| - 5.1.2 - Modelagem de Dados |
| - 5.1.3 - Consultas |
| - 5.1.4 - Revisão |
|  |
| **- 5.2 - Spring Framework** |
| - 5.2.1 - Introdução e Maven |
| - 5.2.2 - JDBC |
| - 5.2.3 - Spring Data JPA |
| - 5.2.4 - Revisão |
|  |
| 6 | Web Applications e Testes | **- 6.1 - Web Applications** |
| - 6.1.1 - Spring Boot |
| - 6.1.2 - Web Services REST com Spring |
| - 6.1.3 - Spring Security (JWT) |
| - 6.1.4 - Integração Angular com Spring |
| - 6.1.5 - Deploy de Aplicações |
| - 6.1.6 - Revisão |
|  |
| **- 6.2 - Teste de aplicações e controle de qualidade** |
| - 6.2.1 - A qualidade das aplicações |
| - 6.2.2 - A qualidade das aplicações e testes |
| - 6.2.3 - Testes unitários em Java |
| - 6.2.4 - TDD e Java |
| - 6.2.5 - Testes comportamentais em Java |
| - 6.2.6 - Revisão |