Atividades Aula 1

**Atividade 1**

**Google: toda a internet é um malware**

Em 2009, um programador cometeu um erro simples, mas que logo tomou grandes proporções. Ao adicionar uma barra invertida em todas as URLs que eram direcionadas ao buscador da empresa, todos os sites listados no Google foram considerados “inseguros”, o que resultava em exibir uma mensagem que notificava o usuário de que um endereço poderia contaminar o seu computador com um malware. Segundo comunicado da empresa, liberado duas horas após o incidente, a situação realmente foi ocasionada por um erro humano, e comprometeu a navegação por apenas 40 minutos.

<https://www.people.com.br/noticias/tecnologia/bugs-os-maiores-erros-da-historia-da-tecnologia>

Questões

* O programador, equipe de desenvolvimento
* Teste em um ambiente controlado antes de mudar o código principal
* Na época em questão a qualidade era media
* Fator humano causando erros
* Milhões de pessoas sem acesso ao serviço, empresas pararam de trabalhar durante o problema

Alguns outros casos de erros

**Worm: primeiro vírus para computadores**

<https://pt.wikipedia.org/wiki/Worm#:~:text=Em%201971%2C%20o%20primeiro%20Worm,entre%20computadores%20conectados%20na%20ARPANET>.

Outros

<https://acervolima.com/10-bugs-famosos-no-mundo-da-ciencia-da-computacao/>

**1. Ariane 5 Rocket Failure (1996): A** explosão do Ariane 5 foi um dos erros de software mais caros da história. Este famoso foguete europeu é usado para lançar um satélite e o custo de desenvolvimento foi de cerca de *8 bilhões* . O foguete explodiu logo após 40 segundos de seu lançamento. A razão por trás de sua falha foi Integer Overflow, que é um bug muito comum na programação de [computadores](https://acervolima.com/10-bugs-famosos-no-mundo-da-ciencia-da-computacao/). É muito importante cuidar do tamanho da memória exigida por uma variável ao declará-la e esse foi o erro durante o desenvolvimento deste foguete. A codificação foi feita em *Ada* . Após a investigação da equipe, eles descobriram que a linha de código tenta ajustar um número de 64 bits em um espaço de 16 bits. O resultado dessa falha foi a perda de US $370 milhões.

**2. Bug Y2K (1999):**Durante a década de 1960, os engenheiros de computação seguiram a abordagem de escrever programas usando um código de dois dígitos para o ano, deixando “19”. A maioria deles achava que escrever “19” antes da variável “ano” era um desperdício de memória desnecessário. Por exemplo, para representar 1970, eles usaram apenas dois dígitos finais “70”. Tudo estava bem até 31 de dezembro de 1999. Mas depois disso, assim que 1 ° de janeiro de 2000, muitos computadores leram o ano como 1900 devido à representação de dois dígitos “00”. Este bug foi chamado de Y2K ou “bug do milênio” (a letra K significa quilo usada para representar o número 1, 000. Portanto, Y2K significa o ano 2000). Afetou bancos que calculam taxas de juros no dia a dia, centros de tecnologia como usinas de energia, transporte e muito mais. Bilhões de dólares foram gastos para atualizar sistemas de [computador](https://acervolima.com/10-bugs-famosos-no-mundo-da-ciencia-da-computacao/) em todo o mundo.

**3. Falha do Míssil Patriota (1991):**Quando um erro é cometido em uma operação militar, a vida de muitos é colocada em risco. Durante a Primeira Guerra do Golfo, um Míssil Patriota americano foi implantado para detectar e interceptar um ataque de míssil Scud que dependia de um tempo muito preciso. Devido ao erro de arredondamento no sistema, ele calculou o tempo incorreto e o míssil não conseguiu interceptar o ataque do míssil Scud iraquiano a um quartel do exército na Arábia Saudita. 28 soldados americanos perderam a vida e 100 pessoas ficaram feridas naquele ataque.

**4. AT&T (1990):**AT&T optou por atualizar um [software](https://acervolima.com/10-bugs-famosos-no-mundo-da-ciencia-da-computacao/) mais complexo do que o atual e espera melhorar as chamadas de longa distância, mas o tiro saiu pela culatra e a rede foi desativada. Por nove horas, os clientes da AT&T não conseguiram fazer uma chamada de longa distância. 75 milhões de ligações perdidas e 200 mil reservas aéreas perdidas devido a esta falha. O problema era uma condição de corrida na troca de código de estação que lida com travamentos de estação de troca. Uma condição de corrida foi causada por uma instrução switch dentro de um loop. A AT&T perdeu US $60 milhões por causa desse erro no código.

**5. Gangnam Style Broke Youtube:** Quando o youtube foi desenvolvido pela primeira vez, ninguém pensava que uma música pode obter bilhões de visualizações e pode exceder o tamanho máximo de um inteiro assinado de 32 bits. O valor máximo para um número inteiro assinado de 32 bits é 2, 147, 483, 647 e quando uma música Gangnam Style chegou às visualizações nesta canção de sucesso da pop star coreana excedeu o valor máximo e obtivemos o resultado abaixo ...

**Atividade 2**

Quais são as características comuns a esses softwares?

Todos os softwares apresentaram falhas e em sua grande maioria por falta de testes

Quais aspectos definem as questões de qualidade?

Na minha opinião a questão da qualidade tem muito a ver com os requisitos do projeto se um software atende a todos os requisitos da proposta e está de acordo com as leis então ele possui qualidade

O que poderia ser feito para melhorar?

Uma boa analise dos requisitos do usuário pode ajudar na qualidade do software e acima de tudo sempre testar antes de apresentar o projeto para o público alvo

Onde os pontos falhos poderiam ser corrigidos?

Poderiam ser feitas auditorias internas e externas com a equipe de desenvolvimento para analisar o software e evitar erros tão graves

**Atividade 3**

Compare as visões de Pressman e Hirama sobre Qualidade de

Software. O que há de comum? O que há de diferente?

Stephen Pressman foca na qualidade técnica do software, como atender aos requisitos, desempenho e usabilidade. Ele acredita que a qualidade é alcançada através de práticas sistemáticas durante o desenvolvimento.

Shinya Hirama vê a qualidade de software como algo que pode ser gerenciado e melhorado através de processos e práticas de gestão. Ele enfatiza a importância do controle de processos e da melhoria contínua.

Semelhanças: Ambos reconhecem a importância de atender às expectativas do cliente e a relevância dos processos para a qualidade do software.

Diferenças: Pressman adota uma abordagem técnica e ampla, enquanto Hirama se concentra mais na gestão e no controle dos processos de desenvolvimento.

Tente definir com suas palavras o conceito de Qualidade de

Software com base no que foi visto;

A qualidade tem muito a ver com os requisitos do projeto se um software atende a todos os requisitos da proposta e está de acordo com as leis além dos outros pontos essenciais como:

Experiência do Usuário (UX): O software deve ser fácil e agradável de usar.

Desempenho e Escalabilidade: Precisa funcionar bem e suportar um aumento de usuários ou dados.

Manutenção e Suporte: Deve ser fácil de atualizar e corrigir problemas.

Segurança: Deve proteger os dados e manter a privacidade dos usuários.

Testes e Validação: Testar bem para garantir que tudo funcione como esperado.

Feedback dos Usuários: Ouvir e responder ao que os usuários dizem pode melhorar a qualidade.

Na prática, como os aspectos vistos podem contribuir para:

1. Avaliar a qualidade de um software existente?

Experiência do Usuário (UX) como os usuários interagem com o software. Testes de usabilidade e feedback dos usuários podem ajudar a identificar áreas que precisam de melhorias na interface e na experiência geral.

2. Construir um novo software com qualidade?

A equipe deve compreender todos os requisitos das partes interessadas e testar funcionalidades a cada ciclo de desenvolvimento para garantir que está alinhada com o objetivo principal

**Atividade 4**

1. Como as normas definem os aspectos de qualidade enumerados anteriormente?

ISO/IEC 9126 (1991):

Funcionalidade: Avaliada pela capacidade do software de atender às necessidades e requisitos especificados.

Confiabilidade: Medida pela estabilidade e confiança do software em operar corretamente sob condições específicas.

Usabilidade: Avaliada pela facilidade com que o usuário pode aprender e usar o software.

Eficiência: Examina o uso de recursos como tempo de resposta e uso de memória.

Manutenibilidade: Refere-se à facilidade com que o software pode ser modificado para corrigir defeitos ou melhorar o desempenho.

Portabilidade: Mede a capacidade do software de ser transferido para diferentes ambientes ou plataformas.

IEC 25010 (2011):

Produto:

Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade: Definidos como antes, mas com acréscimos.

Eficiência de Desempenho: Foca no desempenho do software, incluindo a resposta e uso eficiente dos recursos.

Segurança: Avalia a proteção do software contra ameaças e vulnerabilidades.

Suportabilidade: Examina a facilidade com que o software pode ser suportado e mantido.

Uso:

Eficácia: A capacidade do software de alcançar os objetivos desejados pelos usuários.

Eficiência: A relação entre o desempenho do software e os recursos utilizados.

Satisfação: A percepção do usuário sobre a qualidade e a experiência de uso.

Liberdade de Risco: A medida em que o software minimiza riscos para os usuários.

Cobertura do Contexto (Completude e Flexibilidade): Avalia a capacidade do software de se adaptar a diferentes contextos e suas funcionalidades abrangentes.

Família SQuaRE (atual):

ISO/IEC 25002 (2024): Define modelos de qualidade e fornece uma base para avaliação e gestão da qualidade do software.

ISO/IEC 25010 (2023): Inclui novas características como Capacidade de Interação, Segurança e Flexibilidade, refinando e expandindo as definições anteriores.

ISO/IEC 25019 (2023): Aborda a qualidade no uso, focando em aspectos como Eficácia, Eficiência e Satisfação dos usuários, melhorando a compreensão de como o software é utilizado na prática.

**Atividade 5**

**Tabela**

|  |  |  |
| --- | --- | --- |
| **Português** | **Inglês** | **Definição** |
| **Funcionalidade** | **Functionality** | **A capacidade do software de cumprir os requisitos e fornecer as funções desejadas.** |
| **Confiabilidade** | **Reliability** | **A capacidade do software de manter seu desempenho sob condições específicas e durante o tempo.** |
| **Usabilidade** | **Usability** | **A facilidade com que os usuários podem aprender e usar o software.** |
| **Eficiência** | **Efficiency** | **O uso eficiente dos recursos do sistema, como tempo de resposta e uso da memória.** |
| **Manutenibilidade** | **Maintainability** | **A facilidade com que o software pode ser modificado para correção de defeitos ou melhorias.** |
| **Portabilidade** | **Portability** | **A capacidade do software de ser transferido para diferentes ambientes ou plataformas.** |
| **Eficiência de Desempenho** | **Performance Efficiency** | **O desempenho do software em termos de uso de recursos e tempo de resposta.** |
| **Segurança** | **Security** | **A proteção do software contra ameaças e vulnerabilidades.** |
| **Suportabilidade** | **Supportability** | **A facilidade com que o software pode ser suportado e mantido.** |
| **Eficácia** | **Effectiveness** | **A capacidade do software de alcançar os objetivos desejados pelos usuários.** |
| **Liberdade de Risco** | **Risk Freedom** | A medida em que o software minimiza riscos para os usuários. |
| **Cobertura do Contexto** | |  | | --- | | **Context Coverage** |  |  | | --- | |  | | A capacidade do software de atender a diferentes contextos e sua flexibilidade e completude. |
|  |  |  |
|  |  |  |

Considerando as diferentes versões de normas, houveram Mudanças nessas dentições? São significativas?

Sim, houve mudanças significativas nas definições entre as versões das normas:

* **ISO/IEC 9126 (1991)** para **IEC 25010 (2011)**: A versão mais recente expandiu as características de qualidade, incluindo novos aspectos como **Eficiência de Desempenho**, **Segurança** e **Suportabilidade**.
* **ISO/IEC 25010 (2023)**: Refinou e atualizou as definições, adicionando **Capacidade de Interação**, **Segurança** e **Flexibilidade** para melhor refletir as necessidades e desafios modernos no desenvolvimento de software.
* **Família SQuaRE**: A introdução de normas específicas, como **ISO/IEC 25019 (2023)**, demonstra uma ênfase crescente na qualidade do software no uso real, com foco em **Eficácia**, **Eficiência** e **Satisfação** dos usuários.

Essas mudanças são significativas porque refletem a evolução das tecnologias e práticas de software, abordando novas demandas e desafios na área.

Considerando as normas mais recentes, comente por que há diferentes categorias de normas dentro da Família SQuaRE

A Família SQuaRE é dividida em categorias para abordar diferentes aspectos da qualidade de software de forma mais detalhada:

* **Requisitos:** Definem o que é necessário para que o software atenda às expectativas e necessidades dos usuários.
* **Modelos:** Fornecem uma estrutura e uma visão geral dos atributos e características de qualidade.
* **Gerenciamento:** Abordam a gestão e controle da qualidade ao longo do ciclo de vida do software.
* **Medições:** Definem métricas e métodos para avaliar a qualidade do software de forma objetiva.
* **Avaliação:** Oferecem diretrizes para a avaliação e análise da qualidade do software em diferentes contextos.

Essa divisão permite uma abordagem mais completa e organizada para garantir que todos os aspectos da qualidade do software sejam abordados e gerenciados eficazmente.

**Atividade 6**

1. O que representam os níveis 1 ao 5 no CMMI? O que isso implica para as empresas?

O Capability Maturity Model Integration (CMMI) é um modelo para a melhoria de processos que avalia a capacidade e maturidade de processos de desenvolvimento e gerenciamento em organizações. Os níveis de maturidade no CMMI são classificados de 1 a 5 e representam diferentes estágios de desenvolvimento e capacidade dos processos de uma organização.

**Níveis do CMMI**

1. **Nível 1 - Inicial (Initial):**
   * **Descrição:** Processos são ad hoc e desorganizados. A capacidade de gerenciar projetos é instável e dependente do esforço individual.
   * **Implicações:** Empresas em nível 1 têm práticas de processo não documentadas e muitas vezes enfrentam problemas com a previsibilidade e consistência dos resultados. O sucesso depende fortemente de pessoas individuais e não de processos estabelecidos.
2. **Nível 2 - Gerenciado (Managed):**
   * **Descrição:** Processos são projetados e executados de forma que os resultados são gerenciáveis e previsíveis. As práticas são documentadas e são usadas para planejar, executar e controlar projetos.
   * **Implicações:** Empresas em nível 2 têm processos básicos para gerenciar e controlar projetos, melhorando a previsibilidade e a capacidade de repetição de processos. Há uma ênfase na documentação e controle de processos.
3. **Nível 3 - Definido (Defined):**
   * **Descrição:** Processos são bem definidos e documentados. Eles são padronizados em toda a organização e incluem melhores práticas e técnicas.
   * **Implicações:** Empresas em nível 3 têm processos estabelecidos que são definidos e implementados de forma consistente em toda a organização. Há um foco na documentação, melhoria contínua e treinamento para garantir que os processos sejam seguidos.
4. **Nível 4 - Quantitativamente Gerenciado (Quantitatively Managed):**
   * **Descrição:** Processos são controlados e gerenciados usando métricas e dados quantitativos. A organização monitora o desempenho dos processos e usa dados para entender e melhorar o processo.
   * **Implicações:** Empresas em nível 4 utilizam métricas e análises para entender o desempenho dos processos e prever futuros resultados. Há um foco na coleta e análise de dados para controlar e melhorar a qualidade e eficiência dos processos.
5. **Nível 5 - Em Otimização (Optimizing):**
   * **Descrição:** A organização se concentra na melhoria contínua dos processos. A ênfase está na inovação e otimização dos processos para melhorar o desempenho e adaptar-se a mudanças.
   * **Implicações:** Empresas em nível 5 são altamente adaptáveis e focadas na melhoria contínua. Elas usam dados de desempenho para impulsionar inovações e melhorias de processo. O objetivo é minimizar variações e melhorar continuamente os processos.
6. Responda a questão anterior, considerando o modelo MPS-BR

O Modelo de Processo de Software Brasileiro (MPS-BR) é um modelo de maturidade e melhoria de processos desenvolvido para promover a qualidade do software em organizações no Brasil. Semelhante ao CMMI, o MPS-BR é estruturado em níveis de maturidade que ajudam as empresas a melhorar seus processos de desenvolvimento e gestão de software.

**Níveis do MPS-BR**

1. **Nível G - Inicial (G):**
   * **Descrição:** Neste nível, os processos de desenvolvimento de software são informais e não estruturados. O trabalho é frequentemente realizado de forma ad hoc, e o sucesso do projeto depende fortemente dos esforços individuais e habilidades das pessoas envolvidas.
   * **Implicações:** Empresas no nível G enfrentam desafios com a previsibilidade e a consistência dos resultados. Processos não documentados podem levar a problemas de qualidade e gerenciamento de projeto.
2. **Nível F - Gerenciado (F):**
   * **Descrição:** Os processos são planejados e executados de forma que os resultados possam ser gerenciados e controlados. Há um foco em estabelecer e seguir práticas de processo básicas, e a documentação é usada para garantir que os processos sejam repetíveis.
   * **Implicações:** Empresas no nível F têm uma estrutura básica para gerenciar e controlar projetos. A documentação e o controle de processos ajudam a melhorar a previsibilidade e a consistência dos resultados.
3. **Nível E - Definido (E):**
   * **Descrição:** Os processos são bem definidos, documentados e padronizados em toda a organização. A ênfase está em melhorar continuamente os processos e garantir que eles sejam seguidos de forma consistente.
   * **Implicações:** Empresas no nível E possuem processos organizacionais bem estabelecidos, permitindo uma abordagem mais sistemática para a gestão de projetos e qualidade. A consistência e a melhoria contínua são prioridades.
4. **Nível D - Quantitativamente Gerenciado (D):**
   * **Descrição:** Os processos são gerenciados e controlados usando métricas e dados quantitativos. A organização coleta e analisa dados para entender o desempenho dos processos e fazer ajustes conforme necessário.
   * **Implicações:** Empresas no nível D usam dados e métricas para melhorar o desempenho dos processos. A análise quantitativa ajuda a identificar áreas de melhoria e controlar a variabilidade dos resultados.
5. **Nível C - Em Otimização (C):**
   * **Descrição:** A organização se concentra na melhoria contínua dos processos e na inovação. Há um foco em otimizar os processos e adaptar-se rapidamente às mudanças e novas necessidades.
   * **Implicações:** Empresas no nível C são altamente adaptáveis e voltadas para a inovação. Elas utilizam dados e feedback para melhorar continuamente e se adaptar às mudanças no mercado e nas necessidades dos clientes.
6. Como uma empresa pode se capacitar para usar um modelo de maturidade? Quais são os benefícios disso?

Para uma empresa se capacitar a usar um modelo de maturidade, como o CMMI ou o MPS-BR, ela precisa adotar uma série de práticas e passos estratégicos

1. **Treinamento e Formação:**
   * **Educação e Treinamento:** A equipe precisa ser treinada sobre os conceitos e práticas do modelo de maturidade. Isso pode incluir cursos formais, workshops e treinamentos especializados oferecidos por consultores ou entidades certificadoras.
   * **Certificação:** Profissionais podem buscar certificações específicas que comprovem sua competência na aplicação do modelo.
2. **Avaliação Inicial:**
   * **Diagnóstico de Processos:** Realizar uma avaliação inicial para entender o estado atual dos processos da empresa em relação ao modelo de maturidade. Isso ajuda a identificar lacunas e áreas que precisam de melhoria.
3. **Planejamento e Implementação:**
   * **Desenvolver um Plano de Melhoria:** Com base na avaliação inicial, criar um plano de ação para implementar melhorias nos processos. O plano deve incluir metas, recursos necessários e um cronograma.
   * **Implementar Mudanças:** Aplicar mudanças nos processos de acordo com o modelo de maturidade. Isso pode envolver a documentação de processos, a introdução de novas ferramentas e técnicas, e a reorganização de equipes.
4. **Monitoramento e Avaliação:**
   * **Monitorar Progresso:** Acompanhar o progresso da implementação e fazer ajustes conforme necessário. Utilizar métricas e indicadores para avaliar o desempenho dos processos.
   * **Auditorias Internas:** Realizar auditorias internas para verificar a conformidade com o modelo e identificar oportunidades para melhorias adicionais.
5. **Cultura de Melhoria Contínua:**
   * **Promover uma Cultura de Qualidade:** Incentivar uma cultura organizacional que valorize a melhoria contínua, a documentação e o cumprimento dos processos. Envolver todos os níveis da organização no processo de mudança.
6. **Consultoria e Apoio Externo:**
   * **Contratar Consultores:** Considerar a contratação de consultores especializados que possam ajudar a implementar o modelo de maturidade e orientar a empresa durante o processo de transformação.

**Benefícios de Usar um Modelo de Maturidade**

1. **Melhoria da Qualidade:**
   * **Processos Melhorados:** Adoção de práticas de processo bem definidas resulta em produtos e serviços de maior qualidade. Menos defeitos e maior conformidade com os requisitos do cliente.
2. **Maior Previsibilidade e Controle:**
   * **Previsibilidade de Resultados:** Processos maduros permitem uma melhor previsibilidade dos resultados e a capacidade de gerenciar projetos de forma mais eficiente.
   * **Controle de Processos:** Melhoria na capacidade de monitorar e controlar o desempenho dos processos, identificando e corrigindo problemas antes que eles se agravem.
3. **Eficiência Operacional:**
   * **Redução de Desperdícios:** Processos mais eficientes e bem geridos ajudam a reduzir desperdícios e melhorar o uso dos recursos.
   * **Melhoria na Produtividade:** Maior eficiência nos processos pode levar a um aumento na produtividade da equipe e na entrega de projetos.
4. **Gerenciamento de Riscos:**
   * **Identificação e Mitigação de Riscos:** Melhoria na capacidade de identificar e mitigar riscos, resultando em menos surpresas e problemas durante o ciclo de vida do projeto.
5. **Vantagem Competitiva:**
   * **Diferenciação no Mercado:** Empresas com processos bem definidos e de alta qualidade têm uma vantagem competitiva, oferecendo produtos e serviços superiores.
   * **Reconhecimento:** A certificação no modelo de maturidade pode melhorar a reputação da empresa e facilitar a conquista de novos clientes e contratos.
6. **Cultura de Melhoria Contínua:**
   * **Inovação e Adaptação:** Promoção de uma cultura de melhoria contínua e inovação, ajudando a empresa a se adaptar rapidamente às mudanças e evoluir conforme as necessidades do mercado.

4. Pesquise por empresas que sejam certificadas em algum modelo de maturidade.

Empresas Certificadas em CMMI (Capability Maturity Model Integration)

IBM (International Business Machines Corporation):

Cognizant

TOTVS: Certificação: MPS-BR Nível G e F

SAS (Soluções em Automação e Sistemas): Certificação: MPS-BR Nível E

Resource IT Solutions: Certificação: MPS-BR Nível E

**Atividade 7**

**Institute of Electrical and Electronics Engineers (IEEE)**

* Tecnologia e Inovação: O IEEE foca em promover a inovação tecnológica em várias áreas da engenharia elétrica, eletrônica e computação. Isso inclui pesquisas em inteligência artificial, redes, sistemas embutidos, telecomunicações e muito mais.
* Padrões e Normas Técnicas: O IEEE é conhecido pela criação de normas técnicas internacionais que orientam o desenvolvimento de tecnologias, incluindo padrões para redes sem fio (como Wi-Fi), software, hardware, entre outros.
* Educação e Capacitação: A IEEE oferece recursos educacionais, como cursos, webinars, publicações, e conferências, visando capacitar profissionais e estudantes nas áreas de engenharia e tecnologia.
* Publicações e Revistas Científicas: O IEEE publica uma vasta gama de periódicos, revistas, e conferências científicas que abordam avanços nas áreas de engenharia elétrica e eletrônica, sendo uma das principais fontes de referência para pesquisas acadêmicas e industriais.
* Conferências e Eventos: A IEEE organiza diversas conferências e eventos internacionais onde pesquisadores, engenheiros e acadêmicos discutem as últimas tendências e avanços tecnológicos.

**Association for Computing Machinery (ACM)**

* Ciência da Computação e Engenharia de Software: A ACM cobre uma ampla gama de tópicos relacionados à ciência da computação, incluindo algoritmos, arquitetura de computadores, inteligência artificial, engenharia de software, e mais.
* Publicações Acadêmicas: A ACM é conhecida por suas revistas e conferências altamente respeitadas, como a "Communications of the ACM" e "ACM SIGGRAPH", que publicam pesquisas de ponta em computação.
* Educação em Computação: A ACM oferece materiais educacionais e currículos para apoiar a educação em ciência da computação em diferentes níveis, desde o básico até o avançado.
* Grupos de Interesse Especial (SIGs): A ACM tem vários Grupos de Interesse Especial (SIGs) que permitem a profissionais e acadêmicos focarem em subáreas específicas da computação, como SIGCHI (Interação Humano-Computador) e SIGPLAN (Linguagens de Programação).
* Eventos e Conferências: A ACM organiza uma série de conferências internacionais focadas em diferentes áreas da computação, permitindo que profissionais compartilhem conhecimentos e explorem novas tecnologias.

**Sociedade Brasileira de Computação (SBC)**

* Educação e Formação em Computação: A SBC é muito ativa na promoção da educação em ciência da computação no Brasil, desenvolvendo diretrizes curriculares, materiais didáticos, e eventos voltados para professores e alunos.
* Pesquisa e Desenvolvimento: A SBC apoia e promove a pesquisa em várias áreas da computação, como bancos de dados, inteligência artificial, sistemas distribuídos, e mais, por meio de eventos, publicações e grupos de trabalho.
* Inclusão Digital e Social: A SBC promove iniciativas voltadas para a inclusão digital, buscando aumentar o acesso à tecnologia e ao ensino de computação em comunidades menos favorecidas.
* Conferências e Eventos: A SBC organiza uma série de eventos e congressos anuais, como o Congresso da Sociedade Brasileira de Computação (CSBC), que reúne acadêmicos, profissionais e estudantes para discutir os avanços e desafios da área.
* Divulgação Científica e Publicações: A SBC publica periódicos e revistas que cobrem diferentes aspectos da ciência da computação e promove a disseminação do conhecimento científico na área.

Artigos :