



UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE LETRAS
DEPARTAMENTO DE LÍNGUAS ESTRANGEIRAS E TRADUÇÃO
LETRAS - TRADUÇÃO - FRANCÊS

MARIANA BERNARDO VIEIRA

***LE CODE SOURCE: PROPOSTA DE UM GLOSSÁRIO BILÍNGUE ONLINE
FRANCÊS-PORTUGUÊS BRASILEIRO SOBRE PRINCÍPIOS DA
PROGRAMAÇÃO***

BRASÍLIA, DF
Setembro, 2022

MARIANA BERNARDO VIEIRA

***LE CODE SOURCE: PROPOSTA DE UM GLOSSÁRIO BILÍNGUE ONLINE
FRANCÊS-PORTUGUÊS BRASILEIRO SOBRE PRINCÍPIOS DA
PROGRAMAÇÃO***

Trabalho de Conclusão de Curso apresentado ao Departamento de Línguas Estrangeiras e Tradução como requisito parcial para a obtenção do título de Bacharel em Letras - Tradução - Francês.

Orientador: Prof. Dr. Jean-Claude Miroir.

BRASÍLIA, DF

2022

MARIANA BERNARDO VIEIRA

***LE CODE SOURCE: PROPOSTA DE UM GLOSSÁRIO BILÍNGUE ONLINE
FRANCÊS-PORTUGUÊS BRASILEIRO SOBRE PRINCÍPIOS DA
PROGRAMAÇÃO***

Trabalho de Conclusão de Curso apresentado
ao Departamento de Línguas Estrangeiras e
Tradução como requisito parcial para a
obtenção do título de Bacharel em Letras -
Tradução - Francês.

Aprovado em 15/09/2022

Prof. Dr. Jean-Claude Miroir (Orientador)

Prof. Dr. Eclair Antônio Almeida Filho (1o. Membro da banca)

Prof. Dra. Flávia Cristina Cruz Lamberti Arraes (2o. Membro da banca)

AGRADECIMENTOS

Primeiramente gostaria de dizer que tive apoio de várias pessoas na minha caminhada nesses cinco anos de graduação. Não teria conseguido sem vocês!

Gostaria de agradecer à Deus por me dar forças e me mostrar o caminho que deveria seguir. Por tempos me senti sem rumo e sem saber se conseguiria conquistar qualquer coisa, bom, aqui estou.

Por vezes a comparação com o próximo abala toda sua caminhada, ou pelo menos, parte dela. O sentimento de ficar para trás – em especial no mundo acadêmico – nos faz pensar nossas escolhas e pode nos instigar a desistir, por isso, agradeço à minha mãe por se colocar ao meu lado e sempre me dar todo apoio e confiança que eu precisava. Ao meu avô por confiar em mim e me assistir durante minhas viagens, em especial meu voluntariado, e ao meu pai que sempre cuidou para que eu tivesse uma boa educação.

Gostaria de agradecer à Universidade de Brasília, por me mostrar que toda ciência é válida e me incentivar a percorrer e experienciar todas as oportunidades possíveis. Foram monitorias, tutorias, congressos, seminários, palestras, estágios e projetos de extensão e iniciação científica. Cada uma dessas experiências contribuiu para que eu me tornasse a pessoa que sou hoje, e isso não tem preço.

Agradeço imensamente à professora Sabine Gorovitz, que viu um potencial em mim quando nos conhecemos, e me convidou a compor o MOBILANG – projeto mais que especial – como extensionista. Esse convite mudou toda minha caminhada e não sei o que teria feito caso não tivesse conhecido o mundo da tradução e interpretação comunitária. Hoje fico tão feliz em ver o NuTrad se expandindo e todo o trabalho que está sendo feito na DPU que não tenho palavras para descrever!

Ao professor Eclair, que por anos foi coordenador do nosso curso e sempre se manteve à disposição para nos ajudar (mesmo em meio aos corredores do LET). Obrigada pelas aulas de textos gerais e econômicos, pelas conversas literárias e pelos papos descontraídos nas manhãs de terças e quintas, mas não podia deixar de exaltar o esforço e a paciência durante o período de matrícula, obrigada por me conceder todas as matérias e aceitar minhas propostas desconcertadas de grade semestral!

Por falar em orientadores, não posso deixar de agradecer à professora Elisa Teixeira, que me aceitou como orientanda do projeto COMMIRE e me ensinou muito (mesmo à distância). Sem a professora Elisa, a Linguística de Corpus nunca teria sido tão bem

inserida na minha vida, e esse trabalho não teria sido possível. Muito obrigada por sempre nos incentivar e ter paciência com meus atrasos (foram por uma boa causa)!

Ao professor Jean Claude Miroir, que me deu outra perspectiva quanto ao TCC, esses últimos meses foram extremamente decisivos e sempre pensei em desistir, muito obrigada por confiar em mim quanto ao meu projeto e sempre me ensinar e mostrar quais caminhos deveria seguir. Peco perdão pelos meus envios às manhãs pouco antes de nossos encontros semanais e agradeço muito (muito mesmo) pela paciência e por tudo que aprendi! Tenho certeza que com sua orientação e um pouco mais de tempo, conseguiria ter feito mais, e espero um dia fazê-lo com o que aprendi.

Por último, gostaria de agradecer à Quimera que me deu o incentivo que precisava para começar minha jornada no mundo profissional. Tudo que conquistei na minha carreira foi graças à experiência que tive na nossa amada (e tão importante) empresa júnior.

LISTA DE FIGURAS

- Figura 1. Imagem de tela de um arquivo pdf compilado no corpus francês.
- Figura 2. Página inicial do projeto OCR do ABBYY FineReader.
- Figura 3. Página principal de projeto no Memsources.
- Figura 4. TB visualizada em formato *.xlsx*.
- Figura 5. Modelo de ficha bilíngue padrão com campos vazios.
- Figura 6. Tela do site TermoStat.
- Figura 7. Tela do relatório de resultados de corpus do TermoStat.
- Figura 8. Ficha preenchida “variable globable-variável global”.
- Figura 9. Tela do HTML básico disponibilizado pelo W3S.
- Figura 10. Imagem de tela de instrução em programação C no livro *Principes de Programmation*.
- Figura 11. Página inicial do *Le code source*.
- Figura 12. Página inicial do *Le code source* no GitHub.
- Figura 13. Página do *Le code source* em formato *.pdf*.
- Figura 14. Página de resultados para “informática” na ferramenta KWIC do AntConc.
- Figura 15. Página de resultados para “computação” na ferramenta KWIC do AntConc.
- Figura 16. Página de resultados para “objet-orientée” na ferramenta KWIC do AntConc.
- Figura 17. Página de resultados para “orientée-objet” na ferramenta KWIC do AntConc.

LISTA DE QUADROS

- Quadro 1. Critérios utilizados para constituição dos corpora.
- Quadro 2. Número de types e tokens nos corpora do projeto.
- Quadro 3. Número de types e tokens nos corpora do projeto após a primeira análise.
- Quadro 4. TB do projeto de tradução visualizada em formato *.xlsx*.

APÊNDICE

- Apêndice I – Fichas UTEs completas do *Le code source*.
- Apêndice II – Lista de UTEs e palavras-chave.
- Apêndice III – Lista de UTEs extraordinárias.
- Apêndice IV - Tradução espelhada de “Principes de Programmation”.

LISTA DE UTES

| | |
|--|----|
| <i>affectation</i> //atribuição | 46 |
| <i>appel de fonction</i> //chamada de função | 46 |
| <i>arbre binaire</i> //árvore binária | 47 |
| <i>assembleur</i> //montador | 47 |
| <i>base de données</i> //banco de dados | 48 |
| <i>booléenne</i> //booleano | 49 |
| <i>boucle</i> //laço | 49 |
| <i>chaîne de caractère</i> //cadeia de caractere | 50 |
| <i>code machine</i> //código de máquina | 51 |
| <i>code source</i> //código fonte | 51 |
| <i>compilateur</i> //compilador | 52 |
| <i>débogueur</i> //depurador | 53 |
| <i>déclaration de variable</i> //declaração de variável | 53 |
| <i>déclaration</i> //declaração | 54 |
| <i>erreur de compilation</i> //erro de compilação | 54 |
| <i>exécuter des tâches</i> //executar tarefas | 55 |
| <i>exécuter un programme</i> //executar um programa | 56 |
| <i>fichier binaire</i> //arquivo binário | 56 |
| <i>fichier d'en-tête</i> //arquivo de cabeçalho | 57 |
| <i>fonction récursive</i> //função recursiva | 58 |
| <i>fonction</i> //função | 58 |
| <i>informatique</i> //computação | 59 |
| <i>instruction d'affectation</i> //instrução de atribuição | 60 |
| <i>instruction en langage machine</i> // instrução em linguagem de máquina | 61 |
| <i>instruction</i> //instrução | 61 |
| <i>langage de bas niveau</i> //linguagem de baixo nível | 62 |
| <i>langage de haut niveau</i> //linguagem de alto nível | 63 |
| <i>mnémonique</i> //mnemônico | 63 |
| <i>opérateur d'affectation</i> //operador de atribuição | 64 |
| <i>opérateur d'incrémentation</i> //operador de incremento | 65 |

| | |
|--|----|
| <i>opérateur//operador</i> | 65 |
| <i>programmation orientée objet//programação orientada a objetos</i> | 66 |
| <i>programmation structurée//programação estruturada</i> | 67 |
| <i>réursive//recursivo</i> | 68 |
| <i>séquence d'échappement//sequência de escape</i> | 69 |
| <i>structure de contrôle//estrutura de controle</i> | 69 |
| <i>structure de données//estrutura de dados</i> | 70 |
| <i>système d'exploitation//sistema operacional</i> | 71 |
| <i>tableau à deux dimensions//matriz bidimensional</i> | 71 |
| <i>tableau//matriz</i> | 72 |
| <i>valeur d'une variable//valor de uma variável</i> | 73 |
| <i>variable globale//variável global</i> | 73 |
| <i>variable locale//variável local</i> | 74 |
| <i>variable//variável</i> | 75 |

SUMÁRIO

| | | |
|-----------|---|-----------|
| 1. | INTRODUÇÃO..... | 13 |
| 1.2 | A CIÊNCIA DA COMPUTAÇÃO COMO ALICERCE DO MUNDO GLOBALIZADO..... | 14 |
| 2. | FUNDAMENTAÇÃO TEÓRICA..... | 15 |
| 2.2. | LINGUÍSTICA DE CORPUS NA ELABORAÇÃO DE GLOSSÁRIOS BILÍNGUES | 16 |
| 2.3. | BANCO DE DADOS TERMINOLÓGICO BILÍNGUE: A OTIMIZAÇÃO DA EXTRAÇÃO E ANÁLISE DOS CORPORA..... | 17 |
| 3. | METODOLOGIA..... | 19 |
| 3.1 | ESTUDO PILOTO..... | 19 |
| 3.1.1 | O ponto de partida: Composição do corpus paralelo do <i>Glossário</i> | 19 |
| 3.2. | PROCESSO DE TRADUÇÃO: UMA EXPERIÊNCIA TERMINOLÓGICA..... | 24 |
| 3.3. | O MODELO DE FICHA PARA COLETA DE UTES..... | 28 |
| 3.4. | COLETA DE DADOS E PREENCHIMENTO DAS FICHAS..... | 26 |
| 3.5. | <i>LE CODE SOURCE</i> , O GLOSSÁRIO BILÍNGUE SOBRE PRINCÍPIOS DA PROGRAMAÇÃO..... | 33 |
| 4. | RESULTADOS..... | 35 |
| 4.1. | TERMOS NOTÁVEIS..... | 38 |
| 5. | CONSIDERAÇÕES FINAIS..... | 42 |
| | REFERÊNCIAS..... | 44 |
| | APÊNDICE..... | 46 |

RESUMO

Este trabalho trata da construção de um glossário terminológico bilíngue francês-português brasileiro no âmbito da programação. Partimos da hipótese que a terminologia técnica deve ser acessível e assim, devidamente reconhecida pelos tradutores para promoção de traduções precisas e adequadas em função dos usos da linguagem especializada de uma determinada área. Em consequência, partimos da conjuntura de uma considerável escassez de materiais e recursos em terminologia bilíngue francês-português brasileiro na área da ciência da computação, o que consequentemente gera uma dificuldade para os tradutores, especialmente para os tradutores que estão se inserindo em tal domínio, durante o processo tradutológico. Tomando por base as teses mencionadas anteriormente e as perspectivas teóricas-metodológicas da Terminologia e da Linguística de Corpus, o presente estudo tem como objetivo promover a produção científica na área da terminologia bilíngue, passando pelos fundamentos da Linguística de Corpus, de forma a registrar e analisar o processo assistido por ferramentas de LC na tradução de um texto técnico, nosso objeto de estudo base. Dessa forma, nosso projeto visa alcançar o público-alvo (tradutores nos pares de língua francês e português brasileiro) da pesquisa e divulgar o produto final aqui desenvolvido, para que especialistas possam continuar a inserir novas entradas, tornando o *glossário* cada vez mais útil ao público-alvo a que se destina.

Palavras-chave: Glossário bilíngue; Linguística de Corpus; Terminologia; Ciência da Computação; Francês-português brasileiro.

ABSTRACT

This research study concerns the construction of a bilingual French-Brazilian Portuguese terminology glossary in the domain of programming. Based on the hypothesis that technical terminology should be accessible and thus properly recognized by translators to promote accurate and appropriate translations for specialized language usage in each field. Accordingly, a considerable scarcity of materials and resources on bilingual French-Brazilian Portuguese terminology in the field of computer science is assumed, which consequently poses a difficulty for translators, especially for those who are entering this field, during the translation process. Based on the above mentioned and on the theoretical-methodological perspectives of Terminology and Corpus Linguistics, this study aims to promote scientific production in the field of bilingual terminology, crossing the fundamentals of specialized (technical) translation, by documenting and analyzing the process assisted by LC tools and proposing translation strategies, in this sense, in terms of reaching the target audience (translators in the French and Brazilian Portuguese language pairs) of this research, and in order to disclose the final product developed hereby, so that specialists may continue to input new entries, making the glossary increasingly useful to its target audience.

Keywords: Bilingual glossary; Corpus Linguistics; Computer Science; French- Brazilian Portuguese.

RÉSUMÉ

Le présent article traite de la construction d'un glossaire terminologique bilingue français-portugais-brésilien en matière de programmation. On part de l'hypothèse que la terminologie technique devrait être accessible et donc, correctement reconnue par les traducteurs pour la promotion de traductions précises et adéquates selon les usages linguistiques spécialisés d'un domaine donné. Par conséquent, nous partons de la conjoncture d'un manque considérable de matériaux et de ressources sur la terminologie bilingue français-brésilien-portugais dans le domaine de l'informatique, ce qui génère par conséquent une difficulté pour les traducteurs, surtout pour ceux qui entrent dans ce domaine, pendant le processus de traduction. Basée sur les thèses mentionnées ci-dessus et sur les perspectives théoriques-méthodologiques de la Terminologie et de la Linguistique de Corpus, la présente étude a pour but de promouvoir la production scientifique dans le domaine de la terminologie bilingue, en passant par les fondements de la traduction spécialisée (technique), dans la mesure où le processus est enregistré et analysé à l'aide des outils LC et où des stratégies de traduction sont proposées, Afin d'atteindre le public cible (les traducteurs des paires de langues française et portugaise brésilienne) de la recherche et dans le cadre de la diffusion du résultat final développé ici, afin que les spécialistes puissent continuer à insérer de nouvelles entrées, rendant le glossaire toujours plus utile à son public cible.

Mots-clés: Glossaire bilingue; Linguistique de Corpus; Informatique; Français; Portugais-Brésilien.

1. INTRODUÇÃO

A industrialização e a chegada da era moderna suscitaram o início de um processo intenso de globalização, um fenômeno que a partir de um conjunto de movimentos socioeconômicos que alteraram a dinâmica mundial, possibilitou o desenvolvimento da chamada sociedade de informação, baseada principalmente na difusão de dados, informação e conhecimento (ZINS, 2007).

A era digital construiu as bases para um desenvolvimento econômico e social sem precedentes, de modo a alavancar oportunidades em diversas áreas por meio da universalidade e acesso à informação e do desenvolvimento de estruturas de inovação tecnológicas, tais como grandes corporações no mercado da tecnologia como as empresas *Microsoft*, *IBM* e *Oracle*, importantes nomes na indústria de inovação e tecnologia.

Em uma sociedade digital, é notável a imprescindibilidade e utilidade por acesso ao conhecimento nas mais diversas indústrias. Com o desenvolvimento desmesurado assistido pela tecnologia, os métodos de trabalho foram aprimorados, tal como a necessidade de uma formação criteriosa de especialistas para atuação nas frentes de desenvolvimento e pesquisa na indústria. Tais especialistas, por sua vez, se envolvem também na elaboração de manuais, guias e no estabelecimento de métodos de ensino em suas áreas, visando acima de tudo, compartilhar o conhecimento e formar novos profissionais, garantindo assim, sua consistência e continuidade.

Em face a esse cenário, a Linguagem Especializada (LSP) foi caracterizada como linguagem essencial para uso dentro de um campo específico do conhecimento e assim, se faz presente e é de conhecimento de especialistas, aprendizes e entusiastas. Ao nos referirmos à disseminação de tal conhecimento, a tradução é inegavelmente uma ferramenta de grande aplicabilidade para garantia de acesso ao conhecimento multilíngue. Dessa maneira, o presente trabalho se baseia na tradução e a terminologia bilíngue como forma de acesso ao conhecimento especializado por meio da disponibilização de referenciais terminológicos bilíngues no domínio da programação, com base nos princípios teóricos apresentados por (TAGNIN; TEIXEIRA, 2012), da metodologia proposta por Furtado (2019) e de acordo com critérios expostos por (BERBER SARDINHA, 2004) para classificar e definir um corpus. Neste trabalho, evidenciaremos e registraremos a influência do processo tradutológico para levantamento terminológico com auxílio de uma CAT Tool e ferramentas de LC utilizadas na extração de dados linguísticos para a criação de um glossário especializado bilíngue.

O interesse no desenvolvimento do presente projeto manifestou-se graças à extensão realizada no projeto *MOBILANG - Migrações e fronteiras no Distrito Federal: a integração linguística como garantia dos direitos humanos* do Instituto de Letras da UnB, liderado pela Prof. Dra. Sabine Gorovitz, que assume um papel fundamental e interdisciplinar, possuindo parcerias institucionais em todo Brasil e atuando em diversas frentes no âmbito da migração e refúgio. O período de extensão foi substancial para o posterior ingresso na pesquisa desenvolvida na área de Linguística de Corpus (LC) e Terminologia, com linha de pesquisa principal a Terminologia da mobilidade, subprojeto do grupo MOBILANG, na qual favoreceu o desenvolvimento de habilidades, assim como o conhecimento terminológico e referencial em LC, contribuindo notoriamente para a escolha da pesquisa aqui a ser desenvolvida e metodologia definida. O exposto projeto engloba áreas distintas do saber: a Tradução, a Terminologia e a Linguística de Corpus, áreas do conhecimento que podem recorrer ao auxílio da tecnologia e dos recursos disponíveis em diferentes plataformas e softwares para fornecer e produzir soluções linguísticas vantajosas à sociedade como um todo.

No que se segue, apresentamos informações básicas sobre a área da ciência da computação, sua importância e a relação entre a tradução e a linguagem técnica para difusão de conhecimento técnico na sociedade digital. Em seguida serão apresentadas noções básicas da fundamentação teórica, com destaque a abordagem direcionada por corpus empregada na extração e compilação de termos, que envolve conceitos da Linguística de Corpus, da Tradução Especializada e da terminologia bilíngue. Adiante, descrevemos a metodologia empregada na elaboração do glossário online e o processo tradutológico do texto principal. Por fim, expomos e discutimos os principais resultados obtidos ao final do trabalho, sugerindo os próximos passos na continuidade do estudo.

1.1. A CIÊNCIA DA COMPUTAÇÃO COMO ALICERCE DO MUNDO GLOBALIZADO

No âmbito da mundialização, a tecnologia contribui ativamente para o desenvolvimento em diversos campos de estudo e da sociedade. As nomeadas linguagens de programação foram desenvolvidas e adaptadas a partir da popularização da internet e dos computadores. Sendo hoje amplamente utilizadas para resolver problemas ou melhorar o desempenho de um sistema, tais linguagens têm um papel fundamental na computação, sendo comumente empregadas no desenvolvimento de serviços nas mais diversas indústrias. O CC, JavaScript, PHP ou Python (BERTOLINI ET AL., 2019), são exemplos notáveis de linguagens de programação de alto nível.

Desde a criação dos primeiros computadores na década de 1940, os quais eram programados na linguagem *Assembly*, também chamada de linguagem de montagem ou linguagem de baixo nível (BERTOLINI et al, 2019), o progressivo crescimento na extensão de dados demandou um desenvolvimento de sistemas mais operáveis e impulsionou o mercado da tecnologia. Como resultado, foram desenvolvidos diversos tipos de tecnologias que permitem o intercâmbio de informações durante os anos.

Os sistemas modernos conduzem a troca de informações e permitem ampliar o conhecimento disponível aos usuários, tal qual é pressuposto em um mundo globalizado. O universo da programação, por sua vez, tem um papel fundamental nessa troca de informações, e se faz presente em peso na era digital com a popularização da Internet na década de 1990 onde as linguagens se moldaram e o desenvolvimento de software tornou-se cada vez mais importante (BERTOLINI et al, 2019).

Com o entusiasmo da indústria, os profissionais e aprendizes buscam o conhecimento técnico necessário para atuarem no mercado. Assim como qualquer área do conhecimento na programação, cada nova atualização ou descoberta requer um estudo para aperfeiçoamento e, posteriormente, sua aplicação e replicação. Em termos de troca de informação e acessibilidade no mundo globalizado, a tradução anda lado a lado com a inovação, estando presente em todos os setores do mercado, uma vez que o avanço significativo na demanda por tradução foi proporcional à realidade da globalização e a integração de sistemas, culturas e informação nos mais diversos idiomas (ORMENESE, 2018). A partir desse pressuposto, apresentamos em seguida nosso estudo e a aplicação da LC como abordagem analítico-terminológica para composição do nosso produto, o *glossário*.

2. FUNDAMENTAÇÃO TEÓRICA

Dado o exposto acima, se faz necessário apresentar a abordagem teórico-metodológica a qual o presente trabalho é embasado. Introduziremos a Linguística de Corpus (LC) brevemente e apresentaremos a Linguagem Especializada (LSP) atrelada à Terminologia para à tradução técnica, visto que este trabalho tem como público-alvo principal os tradutores técnicos na área de programação.

Na perspectiva da tradução especializada, a aplicação da terminologia da área não assegura pontualmente que uma tradução seja bem sucedida (TAGNIN, 2009; TEIXEIRA, 2008; BYRNE, 2006; AZENHA JR, 1999; KORNING ZETHSEN, 1999); dessa forma, o projeto tradutológico depende também dos referenciais de conhecimento e das habilidades

analíticas do tradutor (FISHBACK, 1998:2); contudo, se faz evidente que o acesso a ferramentas e materiais de referência especializados em conjunto com o conhecimento linguístico e especializado do tradutor compõem uma estrutura de peso que coadjuva no processo e sustenta as escolhas de tradução, garantindo assim a consistência e a assertividade do projeto tradutológico.

Com o desenvolvimento da ciência da computação, especialmente da programação, uma pesquisa que tenha como resultado a elaboração de um glossário terminológico bilíngue é um recurso conveniente para o trabalho de tradutores, terminólogos ou entusiastas pelo domínio em questão. Dessa forma, com a progressão das tecnologias e o avanço no desenvolvimento de softwares que podem ser aplicados em diversos segmentos da sociedade, o acesso às informações técnicas se faz imprescindível no âmbito da difusão de conhecimento no mundo globalizado, na qual a tradução é evidentemente presente e se caracteriza por ser uma peça-chave na indústria.

Em virtude da necessidade de difusão do conhecimento e com base no pressuposto de que a tecnologia, em especial, a programação se faz presente na maior parte dos segmentos da sociedade, foi constatado o benefício da criação de um glossário bilíngue (francês-português brasileiro) a partir da análise terminológica do corpus levantado e da tradução assistida por uma CAT tool de um livro didático sobre os fundamentos da programação (HOUNGUE, 2018, p.20-52), visando fornecer aos tradutores especializados maior suporte durante o processo tradutório, e sobretudo ampliar o acesso à Linguagem Especializada na área da computação, garantindo a acessibilidade bilíngue de termos técnicos amplamente usados na área para tradutores especializados dos pares de línguas francês e português.

Na seção a seguir, discutiremos brevemente a correlação entre LC, LSP e Terminologia.

2.1. LINGUÍSTICA DE CORPUS NA ELABORAÇÃO DE GLOSSÁRIOS BILÍNGUES

A Linguística de Corpus (LC) atua como abordagem de análise da linguagem a partir da compilação de textos em formato eletrônico, os nomeados corpora (TEIXEIRA, 2008; TAGNIN, 2015, p. 19; BERBER SARDINHA, 2004, p. 30).

Em paralelo, a nomeada *LSP* (em português, linguagem para fins especiais), por sua vez, também atua face à terminologia, uma vez que a linguagem se faz necessária em indústrias especializadas. A noção de *LSP* remete ao uso da linguagem em áreas específicas do conhecimento, ou seja, a linguagem aplicada a um nível de expertise em um determinado campo do conhecimento. Dessa forma, a *LSP* é definida pelo uso da língua em contextos específicos,

que requerem um estudo de terminologia ou prática e pesquisa em certa área do conhecimento, sendo assim, o uso especializado da língua.

As linguagens especializadas ou linguagens para fins especiais são empregadas na área da tradução em contextos de tradução técnica. Essas linguagens, como mencionado acima, são caracterizadas pela aplicabilidade da terminologia, ou seja, de termos que indicam conceitos de um domínio de especialidade (ANDRADE, 2001, p. 193). Cabré (1999, p. 59) apresenta que as línguas especializadas representam um grupo de subcódigos que podem se superpor à língua geral.

A Terminologia, por sua vez, possui duas vertentes: a disciplina, conceituada pelo estudo de termos especializados, e a terminologia como unidades da língua que possuem significado específico e ocorrência em contextos específicos (BARROS, 2004; KRIEGER e FINATTO, 2004). Contudo, se faz imprescindível o referencial da língua geral, as características de gênero e da tipologia textual nas duas línguas, e a interação e movimento da língua geral com a terminologia especializada (TEIXEIRA 2008).

Nesta perspectiva, apresentaremos a seguir como a Linguística de Corpus atua atrelada à terminologia como abordagem analítica assistida por ferramentas que auxiliam a exploração dos corpora.

2.2. BANCO DE DADOS TERMINOLÓGICO BILÍNGUE: A OTIMIZAÇÃO DA EXTRAÇÃO E ANÁLISE DOS CORPORA

De acordo com (TAGNIN, 2009; McENERY e HARDIE, 2012 apud FURTADO, 2019), os estudos referentes à LC podem ser divididos em: direcionados por corpus ou baseados em corpus. O primeiro é caracterizado por fazer uso das informações extraídas no corpus com auxílio das ferramentas desenvolvidas para tal como base de observação e teorização de fenômenos linguísticos (McENERY e HARDIE, 2012; TAGNIN, 2009 *apud* FURTADO, 2019). O segundo é tido como metodologia, ao usar o corpus como fonte de dados para testar teorias ou hipóteses pré-definidas na área (McENERY e HARDIE, 2012 apud FURTADO, 2019).

A relação entre a terminologia e Linguística de Corpus é congruente, visto que podem atuar em conjunto sob a perspectiva da análise terminológica de corpora. Essa relação é atrelada à construção de dicionários e glossários monolíngues, bilíngues e mesmo multilíngues, como pode ser visto em Furtado (2019), que aponta ainda que a comunicação intra-área, voltada para

especialistas, é considerada no processo, contudo, os conceitos e suas definições são sempre o principal foco de produção (TEIXEIRA, 2008).

Em relação à extração e análise terminológica com o uso de LC para construção de um glossário, partimos da hipótese apresentada por Krieger e Finatto (2004, p. 145) que apontam que “um banco de dados terminológico se apresenta como um sistema de informações interconectadas, armazenado em um computador, visa a atender as necessidades de consulta de um grupo definido de usuários”. Dessa forma, o presente trabalho propõe então tornar acessível e difundir a linguagem especializada bilíngue, evidenciando as relações de equivalência entre as Unidades de Tradução Especializadas, UTEs (TEIXEIRA, 2008) e propondo estratégias de tradução assistida pela CAT tool previamente definida, o Memsource, garantindo assim sua aplicabilidade, e permitindo uma comunicação vantajosa entre tradutores especializados e entre especialistas da área (CRUZ, 2005, p. 28).

Ao observarmos a produtividade e assertividade no processo terminológico podemos nos referir em especial às metodologias e abordagens atuais de Linguística de Corpus e tradução, ocasionadas pelo investimento em softwares que otimizam o trabalho dos tradutores (ORMENESE, 2018). Tais softwares aprimoram o trabalho, reduzem o tempo gasto em uma pesquisa e combinados com a expertise de um linguista garantem a efetividade e a qualidade da análise.

O uso de tais recursos e informações se faz válido a partir do conhecimento técnico do linguista e/ou tradutor, seja no domínio em questão ou com experiência em tais ferramentas. Para a linguagem especializada, o acesso a referências e a uma terminologia é fundamental. Por exemplo, um conjunto de termos estudados por especialistas e terminólogos em uma determinada área facilita o trabalho do tradutor técnico, contudo sua elaboração se faz a partir de uma extensa pesquisa e seleção prévia, mediante o assunto abordado em dado projeto ou área do conhecimento (GALEANI; ASSOLINI, 2022).

O modelo de *Glossário* online que se instalou, tem como objetivo ser acessível ao público-alvo, e como benefício uma maior flexibilidade na visualização dos termos e suas definições, uma vez que as fichas foram criadas em seguindo um modelo-padrão em formato bilíngue e visando a continuação do projeto com uma ampliação do banco de dados terminológico. Neste projeto, tomamos por definição de glossário a proposta por Barros:

Glossário (termo tolerado: dicionário bilíngue, dicionário multilíngue): pode situar-se tanto no nível do sistema como no da(s) norma(s). Sua principal característica é não apresentar definições, mas tão somente

uma lista de unidades lexicais ou terminológicas acompanhadas de seus equivalentes em outras línguas (BARROS 2004, p. 144).

A criação de fichas terminológicas bilíngues faz parte da metodologia usada na pesquisa para criação do glossário. Quanto à parte operacional, optamos por hospedar nosso glossário na plataforma *GitHub*, de fácil acesso aos usuários, visto que não requer login para acesso ao arquivo, contudo, apresentaremos o processo de criação do banco de dados e plataforma online na unidade 3 deste trabalho. A seguir, expomos o processo inicial do trabalho: nosso estudo piloto.

3. METODOLOGIA

Na unidade anterior expusemos a atuação e aplicação de LC atrelada à terminologia para análise de corpus especializado, visando especificamente, a construção de glossários para tradutores técnicos. A coleta do corpus utilizado neste projeto – a que nomeamos *Le code source* – foi realizada no semestre 2022.1, e sua abordagem teórico-metodológica é baseada em Furtado (2019) - inicialmente apresentada em Furtado e Gorovitz (2017), na qual é sistematizada a metodologia de coleta de um corpus multilíngue no âmbito da imigração e refúgio e em Teixeira (2012) onde a LC é apresentada como metodologia para criação de um dicionário de culinária. Nesta seção, apresentamos os critérios e o processo de coleta de corpus, o preparo dos textos e as ferramentas e rotinas empregadas na extração dos dados, antes de passar para a próxima unidade, na qual relatamos a experiência e os resultados do projeto.

3.1. ESTUDO PILOTO

3.1.1. O ponto de partida: composição do corpus comparado do *glossário*

O objetivo do trabalho de seleção e coleta do corpus foi recolher materiais para um levantamento cauteloso de sua terminologia, iniciando com uma análise de conceitos introdutórios em computação, no domínio da programação com base em LC e com o auxílio de ferramentas de análise linguística AntConc para construção do glossário bilíngue.

No âmbito tradutológico, a construção de um glossário bilíngue (francês e português brasileiro) é de grande valia para tradutores e linguistas; dessa forma, o presente estudo deu início após ser observado um baixo número de materiais de referência bilíngues (sem a interferência do inglês), ou seja, sem a tradução indireta – interferência de uma língua *pivot*, em português e francês no domínio da programação. Foi constatado a predominância de materiais em língua inglesa na plataforma de busca do *Google Acadêmico*, utilizado para

hospedar links ou materiais didáticos e técnico-científicos, que viabiliza e direciona resultados específicos que variam entre artigos e teses, materiais instrutivos ou livros nas mais diversas temáticas e idiomas com base nas preferências do usuário: idioma, ano, autor e tipo de publicação. Tal fato remete à importância do trabalho aqui realizado, e evidencia a necessidade do desenvolvimento de recursos linguísticos bilíngues como instrumento de auxílio e consulta de tradutores técnicos.

Em relação à coleta do material que compõe os corpora em francês e português brasileiro, o trabalho de pesquisa e análise foi feito a partir de consultas orientadas à temática principal deste trabalho, na plataforma de busca acadêmica Google Acadêmico. Os corpora do presente projeto são constituídos por materiais didáticos e instrutivos em francês (língua base) e português brasileiro, os dois idiomas do glossário aqui desenvolvido. Desta forma, o conteúdo dos materiais didáticos selecionados é destinado a alunos de cursos em computação, tecnologia e engenharia com ênfase em computação ou tecnologia, como observado na análise comparativa curricular dos semestres iniciais dos cursos de graduação em Ciência da Computação da Universidade Federal de Minas Gerais, Engenharia da Computação da Universidade Federal de Santa Maria e de Tecnologia da Informação da Universidade Federal do Mato Grosso do Sul. Isto posto, foram selecionados livros, cadernos de exercícios e apostilas com tópicos introdutórios em computação e programação para o levantamento e validação terminológica, visando assim, um corpus representativo para análise bilíngue.

Visando realizar buscas mais bem orientadas e filtrar o conteúdo dos resultados, foram estabelecidos os seguintes parâmetros, com base nos critérios identificados por Berber Sardinha (2004):

Quadro 1 - Critérios utilizados para constituição dos corpora.

| Parâmetro | Características |
|-------------------|--|
| Língua | Bilíngue (português brasileiro e francês) |
| Tipologia textual | Materiais didáticos tais como livros, apostilas, guias e cadernos de atividades utilizados em cursos introdutórios de computação |
| Tamanho | médio-grande |

| | |
|----------------------|---|
| Área de conhecimento | Computação; Programação |
| Data de publicação | Antigo (a partir de 1993) |
| Autoria/Fonte | Universidades e instituições de ensino |
| Tipo de Corpus | Comparável (textos naturais originalmente escritos em cada uma das línguas) |

Fonte: Elaboração própria.

O conteúdo dos materiais coletados refere-se aos fundamentos e a pluralidade de problemas em computação, para que o aluno (leitor alvo) tenha domínio dos princípios de funcionamento das máquinas, bem como dos conceitos de lógica, algoritmos e linguagens em programação, propostos para os cursos introdutórios de computação e programação, como mencionados acima. Para composição dos corpora, foram inicialmente coletados um total de 14 arquivos que somavam cerca de 650.000 palavras nos dois idiomas de trabalho do presente trabalho. Contudo, após uma análise inicial dos corpora, foi-se observado a necessidade de incrementar os corpora em número e conteúdo, sobretudo o subcorpus de francês, que apresentou um baixo número de tokens comparado ao de português, como ilustrado abaixo:

Quadro 2 - Número de types e tokens nos corpora do projeto.

| Língua | Types | Tokens |
|----------------|--------|---------|
| Português (BR) | 18.262 | 459.291 |
| Francês | 14.799 | 260.849 |

Fonte: Elaboração própria.

A etapa inicial tratou da conversão dos textos após uma filtragem de conteúdo e língua de origem, tendo em vista que, para seguirmos a proposta deste projeto, teríamos que observar se tais materiais seriam, de fato, textos originais em língua francesa ou língua portuguesa brasileira ou se eram frutos de traduções de textos originários do inglês ou outro idioma, o que caracterizaria, assim, a interferência de uma língua pivô em nossa análise comparativa. Os arquivos, por sua vez, não foram convertidos em conjunto, de forma a facilitar a identificação e rotulação individual.

Os textos foram convertidos de .pdf - formato original - para .txt, visando ocupar menos espaço na memória interna do computador e, por conseguinte, ser compatível com a ferramenta de análise textual usada, o AntConc, detalhada mais adiante na presente seção.

O uso do *ABBYY Fine Reader*, um software de conversão de arquivos, foi definido para efetiva aplicação dos arquivos na ferramenta de LC, o AntConc, e no caso do texto principal, também na CAT Tool (*computer-assisted translation*) definida para o desenvolvimento da versão final da tradução, um elemento importante no fluxo de trabalho de um projeto tradutológico (ZARETSKAYA et al., 2015; SCHNEIDER et al., 2019), como detalhado na seção 3.2, em seguida.

Dessa forma, foram selecionados e adicionados mais arquivos nos corpora, seguindo os mesmos parâmetros inicialmente estabelecidos no presente projeto. De acordo com a classificação proposta por Berber Sardinha (2004, p.26) em relação aos corpora, neste trabalho o corpus em português é classificado como *médio* e o corpus em francês como *médio-grande*. A integralidade dos materiais textuais coletados nas duas línguas após a primeira análise do projeto resultou no seguinte total de palavras distintas, também chamadas de *types*/palavras corridas, ou *tokens*/palavras-ocorrência:

Quadro 3 - Número de types e tokens nos corpora do projeto após a primeira análise.

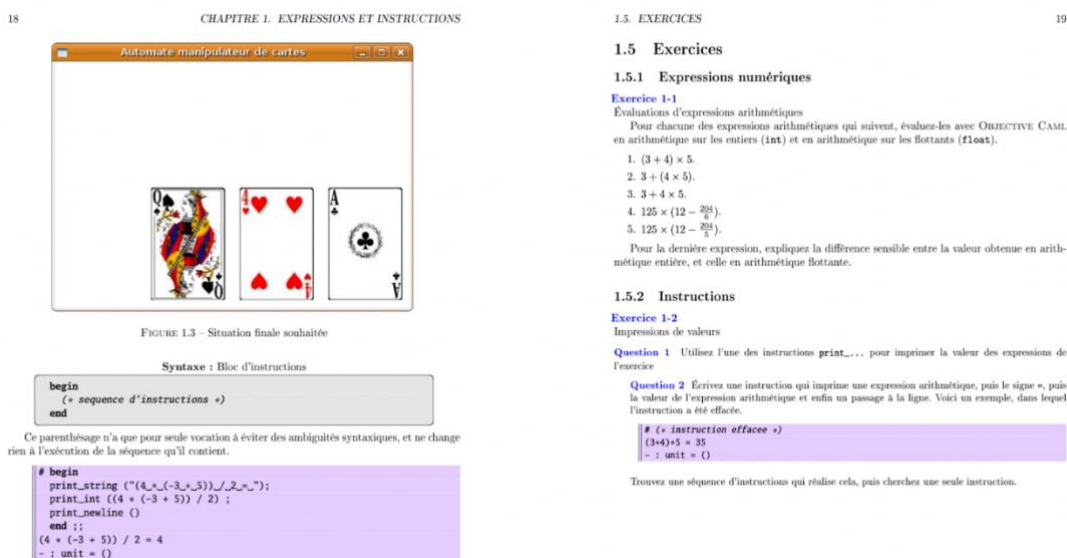
| Língua | Types | Tokens |
|----------------|--------------|---------------|
| Português (BR) | 18.262 | 484.684 |
| Francês | 18.974 | 548.168 |

Fonte: Elaboração própria.

O subcorpus do francês é composto de 14 arquivos de diferentes países francófonos: França, Suíça, Bélgica, Canadá, Argélia, Benin, Senegal, Mauritânia, Burundi e Nigéria. O subcorpus do português brasileiro por sua vez, é composto por 20 arquivos e seu conteúdo foi totalmente produzido no Brasil. Na busca realizada para compilação dos corpora, materiais produzidos pela África francófona foram dados como prioridade inicial, entretanto, devido ao baixo número de resultados, a inclusão desses materiais no corpus foi impossibilitada.

A princípio, artigos científicos também foram considerados para composição do corpus como materiais distintos; contudo, uma análise mais detalhada de seu conteúdo revelou que materiais didáticos abrangem com mais clareza os tópicos introdutórios aqui explorados, além de contarem com definições e exemplos de uso, o que foi de grande valia para a análise linguística comparativa bilíngue, garantindo assim a adição de novos termos no *glossário*, e permitindo o fácil acesso e entendimento de tradutores e linguistas. Os materiais selecionados apresentam macroestruturas de gênero textual semelhantes visto que todos são advindos de instituições de ensino: são compostos por capítulos ou unidades em que apresentam ao leitor uma explicação sobre o conteúdo, e propõem exercícios ou atividades individuais e em grupo sobre o assunto trabalhado. Como exemplo, a figura abaixo ilustra duas páginas de capítulo em um dos arquivos selecionados para compilação do corpus em francês. Uma das páginas apresenta a explicação do capítulo nomeado como *Chapitre 1. Expressions et Instructions* e, em seguida, a seção de exercícios sobre o conteúdo trabalhado, como ilustrado na figura abaixo:

Figura 1: Imagem de tela de um arquivo pdf compilado no corpus francês.



Fonte: Elaboração própria.

Para a análise do corpus, foi definido o uso do software de análise linguística desenvolvido por Laurence Anthony em 2014, o AntConc. Uma vez finalizada a preparação inicial do corpus, pode-se dar prosseguimento ao levantamento e extração terminológica, conforme FURTADO (2019). A seguir, apresentaremos com detalhes o processo tradutológico para o desenvolvimento do glossário.

3.2. PROCESSO DE TRADUÇÃO: UMA EXPERIÊNCIA TERMINOLÓGICA

Para o desenvolvimento do glossário, propusemos a tradução de um dos materiais escolhidos em língua francesa como objeto de tradução. O livro *Principes de Programmation* tem autoria de Pélégie Houngue, professor do Institut de Mathématiques et de Sciences Physiques (IMSP) em Milão, Itália, e foi produzido pela Université Virtuelle Africaine em 2017, consistindo num total de 115 páginas, originalmente em formato .pdf. O arquivo, tal como os demais, foi selecionado e retirado após uma busca no Google Scholar, e é destinado a estudantes e entusiastas em níveis introdutórios da computação; dessa forma, apresenta unidades básicas sobre linguagem de programação e fundamentos da computação. O texto, contudo, apresentou conteúdo condizente com nossa proposta de tradução – macroestrutura, gênero textual, ano de publicação e língua de produção assim, convertido em formato word e revisado para sua utilização na CAT tool definida, o Memsource.

O objetivo da tradução deste texto foi verificar a terminologia para posterior extração e análise do corpora e finalmente, o desenvolvimento do glossário, visando também disponibilizar o arquivo em português brasileiro para demais interessados. Esse processo também consistiu em definir as unidades do texto, dessa forma, foram divididas em três unidades para tradução - *Unité 1: Introduction à la programmation; Unité 2: Unité 2: Résolution de problèmes basée sur l'informatique; Unité 3: Unité 3: Programmation en langage C*, totalizando 64.210 caracteres e/ou 12.126 palavras distribuídas em 61 páginas, incluindo capa e Apresentação. Essa divisão foi realizada com base no conteúdo presente nos arquivos compilados para os corpora de português brasileiro e francês, os quais apresentam capítulos e/ou unidades que abrangem tais conceitos mencionados e visando assim, explorar e analisar a terminologia e confirmar sua consistência.

O uso de ferramentas amplamente utilizadas por tradutores profissionais aumenta a produtividade, melhora a consistência na tradução e, conseqüentemente, reduz o custo da

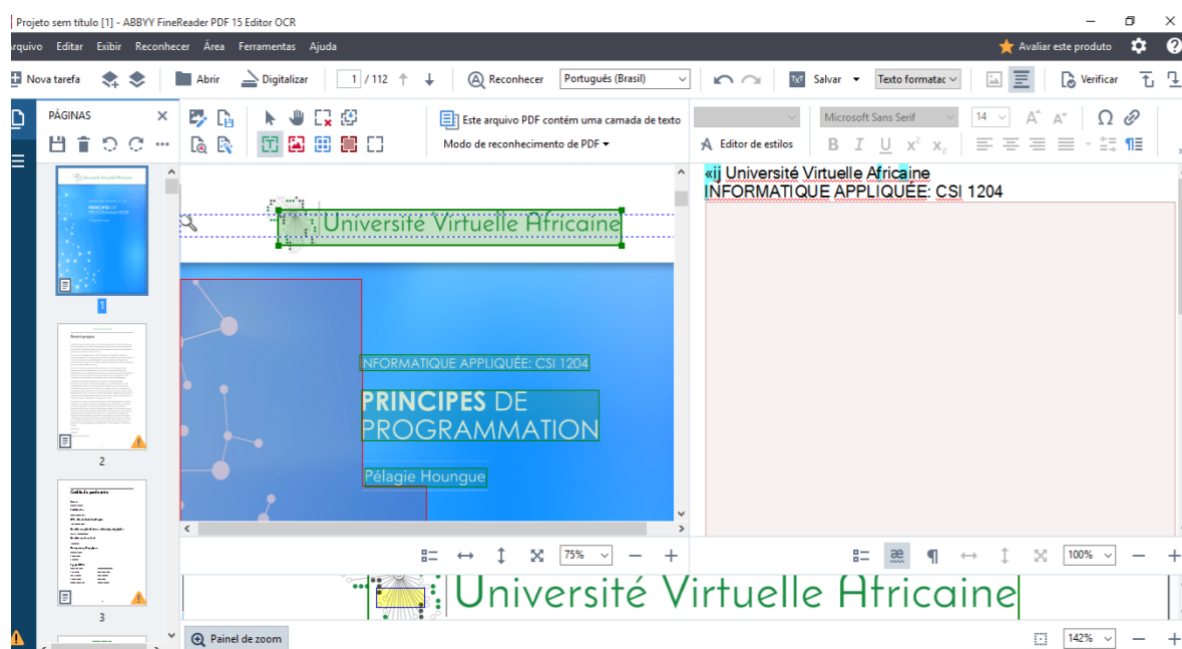
tradução, com destaque as ferramentas CAT e softwares de gerenciamento terminológico (ZAMPIERI E VELA, 2014).

Tais ferramentas apresentam uma variedade de características, dentre as quais é possível citar a TM (memória de tradução), verificação e análise de qualidade (QA), recursos de gerenciamento de glossário e terminologia, sugestões de preenchimento automático e uma série de funcionalidades adicionais relacionadas à logística dos projetos.

Para isso, ao iniciar o processo na ferramenta definida, se fez necessário a conversão do arquivo a ser traduzido para um formato legível (.docx) a fim de ser adicionado sem alterações consideráveis em seu formato ou quaisquer omissões de conteúdo que possam prejudicar o produto. Para essa etapa foi escolhido o *ABBYY FineReader*, uma ferramenta de reconhecimento de caracteres paga e amplamente utilizada em provedores de serviços linguísticos como localização e tradução nas primeiras etapas (preparação) de um projeto de tradução, como previamente utilizada e apresentada no presente projeto. A conversão, edição e revisão do documento foi realizada a partir da criação de um projeto OCR Reconhecimento Óptico de Caracteres (*Optical Character Recognition*) na versão completa da ferramenta ABBYY.

A figura 2 abaixo ilustra a página de projeto OCR no programa ABBYY do documento *Principes de Programmation*. Essa página apresenta a etapa inicial de conversão e revisão de um documento .pdf e os recursos de reconhecimento e edição de elementos textuais disponibilizados pelo programa.

Figura 2: Página inicial do projeto OCR do ABBYY FineReader.



Fonte: Elaboração pr pria.

Em seguida, verificamos a formatação e aspectos gerais e ortográficos do texto, com o objetivo de assegurar uma conversão de qualidade e evitar segmentos e outros erros que possam ocorrer nas etapas seguintes. A verificação textual é um recurso do ABBYY disponibilizado ao usuário uma vez que o reconhecimento de todas as páginas do arquivo no projeto OCR é finalizado. A verificação permite a revisão de figuras e textos delineados e reconhecidos pela ferramenta, facilitando a visualização e edição do texto, bem como a comparação entre arquivo original x arquivo chegada.

Como etapa seguinte, se deu início ao projeto tradutório na CAT Tool definida neste trabalho, o Memsourc, uma ferramenta de tradução paga baseada na nuvem que comporta diversos recursos e oferece aos usuários uma interface intuitiva e fácil gerenciamento de projetos. A ferramenta foi escolhida com base na experiência pessoal por oferecer recursos necessários para criação e gerenciamento do projeto tradutório e apresentar uma interface de nítida compreensão para o usuário. O Memsourc também apresenta uma vantagem evidente em comparação às demais ferramentas, permitindo o compartilhamento de projetos e visualização do progresso em tempo real (ŠANCA, 2018).

Figura 3: Página principal de projeto no Memsourc.

| # | Source: fr | Target: pt-br | |
|----|---|---|---|
| 71 | 1 Objectifs de l'unité 2 | 1 Objetivos da unidade 2 | ✓ |
| 72 | À la fin de cette unité, vous devriez être capable de : | Ao final desta unidade, você deverá ser capaz de: | ✓ |
| 73 | Expliquez ce qu'est la programmation et de définir les concepts de base de la programmation | Explicar o que é programação e definir os conceitos básicos de programação | ✓ |
| 74 | Décrire l'évolution des langages de programmation et leurs influences sur l'évolution de la conception des langages | Descrever a evolução das linguagens de programação e suas influências na evolução do design da linguagem | ✓ |
| 75 | Décrire les niveaux et les générations de langages de programmation | Descrever os níveis e gerações de linguagens de programação | ✓ |
| 76 | Décrire les caractéristiques d'un bon langage | Descrever as características de uma boa linguagem de programação | ✓ |
| 77 | Décrire et expliquer les caractéristiques des paradigmes des langages de programmation | Descrever e explicar as características dos paradigmas da linguagem de programação | ✓ |
| 78 | Principes de Programmation | Princípios de programação | ✓ |
| 79 | Unité 1. | Unidade 1. | ✓ |
| 80 | Introduction à la programmation | Introdução à programação | ✓ |
| 81 | Termes clés | Termos-chave | ✓ |
| 82 | 1 La programmation 1 est l'activité d'écriture d'instructions devant être exécutée par l'ordinateur. | 1 Programação 1 é a atividade de escrever instruções a serem executadas pelo computador. | ✓ |
| 83 | Ces instructions indiquent à l'ordinateur de faire quelque chose. | Essas instruções dizem ao computador para executar uma ação. | ✓ |
| 84 | Par exemple, résoudre un problème particulier, lire les notes des étudiants et calculer le rang par étudiant et de fournir une liste d'étudiants qui passent en classe supérieure et une autre liste des étudiants qui recommenceront le cours. | Por exemplo, resolva um problema específico, leia as notas dos alunos, calcule a classificação por aluno e forneça uma lista de alunos que passarão para a próxima série e outra lista de alunos que repetirão o curso. | ✓ |

Fonte: Elaboração própria.

Para a etapa subsequente, foi realizada uma extensa pesquisa terminológica assistida para início da construção do glossário do projeto. As palavras-chave foram retiradas do projeto de tradução e serviram como base para a posterior pesquisa e análise do corpus. Para validação tradutológica e verificação dos casos de uso, paralelo à construção do glossário.

Após a análise terminológica, a criação da TB (termbase) do projeto foi realizada ao longo da tradução, nos permitindo compilar os termos e adicioná-los na primeira versão do projeto. Essa metodologia foi definida visando a validação da tradução e a aplicação da terminologia com base na revisão da tradução com uso da terminologia extraída do corpus e das observações referenciadas durante o processo. A TB foi salva em formato *.xlsx* para posterior revisão e anexada na própria cat tool para consulta durante a etapa seguinte, o controle de qualidade (QA check). O arquivo é adicionado na aba “glossário” em formato *.xlsx* e conectado ao projeto, detectando assim possíveis inconsistências na fonte ou no texto de chegada, adição ou subtração de itens no segmento, espaços indevidos ou terminologia não reconhecida. O processo de QA garante a qualidade e clareza do texto, e beneficia o tradutor, uma vez que permite que a revisão do texto seja complementada, apontando quaisquer possíveis erros dentro do escopo do projeto que o usuário (tradutor) esteja trabalhando. Ademais, o QA assegura a conformidade e consistência no projeto, visando a qualidade antes da entrega final

Após a conclusão do projeto na plataforma online do Memsource, a TB (*termbase*) anexada ao projeto também teve os termos confirmados, com objetivo de exportar e criar um arquivo de controle (*arquivo_controleptbr*) em *xlsx*. criando assim, uma lista de palavras-chave provenientes do texto de partida, que nos servirão como base para a pesquisa terminológica, como ilustrado na figura abaixo:

Quadro 4: TB do projeto de tradução visualizada em formato *.xlsx*.

| | |
|------------------------|-------------------------|
| informatique | computação |
| instruction | instrução |
| tâches de calcul | tarefas de computação |
| langage de haut niveau | linguagem de alto nível |
| mnémonique | mnemônico |
| instruction de boucle | Instrução de laço |
| instruction de saut | instrução de salto |
| intruction imbriqué | aninhado |
| instruction | instrução |
| déclaration | declaração |
| boucle | loop |
| assembleur | montador |

| | |
|---------------------------|----------------------|
| tableau | matriz |
| opérateur | operador |
| erreur de compilation | erro de compilação |
| commande de débogage | comando de depuração |
| système d'exploitation | sistema operacional |
| environnement appellant | ambiente de chamada |
| variable locale | variável local |
| variable globale | variável global |
| séquence d'échappement | sequência de escape |
| transtypage | tipagem |
| tableau à deux dimensions | matriz bidimensional |
| compilateur | compilador |
| opérateur binaire | operador binário |
| chaîne de caractère | cadeia de caractere |
| variable de contrôle | variável de controle |
| structure de sélection | estrutura de seleção |
| exécuter un programme | executar um programa |
| exécuter l'instruction | executar a instrução |

Fonte: Elaboração própria.

Em suma, o processo de tradução no nosso projeto foi essencial para exploração inicial do corpus, uma vez que serviu como base para o levantamento de palavras-chave que foram o ponto de partida para construção do *Le code source*, ou seja, auxiliaram o processo de extração das UTEs e da criação de fichas terminológicas para compilação do glossário.

3.3. O MODELO DE FICHA PARA COLETA DE UTES

Levando em consideração as premissas da definição de um glossário, apontadas por Barros (2004), bem como as necessidades do nosso público-alvo, definimos um modelo de ficha padrão para as UTEs após extração e análise terminológica.

Figura 4: Modelo de ficha bilíngue padrão com campos vazios.

```

UTE:
fr:
-
pt:
-
UTES_relacionadas:
fr:
-
pt:
-
def:
fr:
-
source:
-
pt:
-
source:
-
contexto:
fr:
-
pt:
-
id:
-
researcher:
-
last_update:
-

```

Fonte: elaboração própria.

A figura acima faz correspondência ao modelo de ficha de UTE, inicialmente sugerido por Furtado (2019), com algumas alterações em sua composição. Nessa ilustração, contabilizamos os seguintes campos:

- UTE - corresponde à UTE extraída e analisada, com uma entrada em francês e uma em português;
- UTEs_relacionadas - faz referência às UTEs com núcleos semelhantes;
- Def – apresenta duas definições extraídas de materiais coletados dos corpora ou outras fontes confiáveis, tais como universidades e instituições de ensino;
- Source - apresenta a referência da fonte verificada em francês
- Fonte - apresenta a referência da fonte verificada em português brasileiro;
- Contexto - apresenta duas sentenças extraídas da ferramenta KWIC do software AntConc, uma em francês e outra em português brasileiro;
- ID - numeração da ficha para controle interno.
- Researcher - apresenta o nome do pesquisador(a) responsável pela criação da ficha;
- Last-update - apresenta a data de última modificação da ficha.

Os campos “Researcher” e “Last-update” foram adicionados em inglês na ficha com base no modelo proposto por Furtado (2019), no qual referencia a necessidade de seguir tal formato para adição na ferramenta App Engine, onde o glossário COMMIRE foi hospedado para acesso dos usuários. Foi observado a importância de seguir com o mesmo formato nas fichas UTE devido à ambição de prosseguir com o presente projeto e expandir o banco de dados,

adicionando a um banco de dados desenvolvido a partir de uma linguagem de programação, que utiliza o inglês como língua padrão.

A seguir, apresentaremos o processo de extração e seleção de dados para o preenchimento e conclusão das fichas UTEs com base nos termos extraídos e analisados.

3.4. COLETA DE DADOS E PREENCHIMENTO DAS FICHAS

Para o nosso projeto de análise e levantamento terminológico, definimos o projeto de tradução como ponto de partida. Após a tradução do texto definido (HOUNGE, 2017), os termos extraídos inicialmente serviram como referencial para o início do trabalho de pesquisa nos corpora. Para o corpus, como primeiro passo, carregamos as listas de palavras na ferramenta “Wordlist” no AntConc nos corpora de francês e português brasileiro, visando adquirir uma visão geral do corpus e conferir aspectos básicos da terminologia. Nesta ferramenta, é possível selecionar uma palavra ou expressão e analisar, a partir desta, suas colocações e contexto de uso; os dois próximos passos, detalhados também nesta seção.

A partir da lista gerada pela ferramenta e pela seleção da palavra, seguimos para a próxima análise: O N-gram. A ferramenta N-Gram no AntConc conduz a uma lista de frequência de sequências da palavra selecionada, permitindo que o usuário tenha acesso ao escopo de expressões e termos relacionados àquela palavra, configurando, assim, uma análise mais detalhada e a descoberta de expressões recorrentes e adição de candidatos a termos. A extração focada no AntConc se refere à análise de contextos de uso nos corpora, ou seja, ao uso da ferramenta KWIC (*concordance*) como meio de verificação de termos. Dessa forma, se faz possível ao usuário analisar e comparar a relação conceito > definição no corpus francês e português brasileiro, auxiliando também o processo de preenchimento do arquivo controle para organização interna e no preenchimento manual das fichas de UTEs, que discutiremos detalhadamente na próxima unidade.

O software AntConc favoreceu considerável parte da presente pesquisa com as ferramentas Wordlist, N-Gram e KWIC; contudo, para análise adicional, utilizamos o *TermoStat Web*, a ferramenta online e gratuita de análise de corpus da Université de Montréal para gerar uma lista de termos no corpus de francês. A ferramenta permite ao usuário escolher os tipos de extração e o idioma, classificando os termos e facilitando a consulta e análise. Dessa forma, temos as seguintes opções no menu inicial de pesquisa na plataforma:

- nomes;

- adjetivos;
- advérbios;
- verbos.

Figura 5: Imagem de tela do site TermoStat.

Fonte: elaboração própria.

A ferramenta ordena a lista de palavras colocando no topo as que possuem maior frequência, especificidade do termo, maior probabilidade de ocorrência, colocações mais típicas, sintaxe e variante ortográfica. A ferramenta também disponibiliza uma lista completa dos termos para que o usuário consiga explorar e analisar os candidatos a termos, bem como expressões relacionadas, colocações típicas e frequências de co-ocorrência de todas as palavras. Uma vez que os resultados são gerados pela ferramenta, o usuário pode exportá-los em formato .txt. No nosso caso, pode-se exportar um tipo de resultado por vez; em nosso caso, exportamos a Liste des termes para o formato .txt; contudo, a visualização prática e a interface dinâmica da ferramenta facilitam a análise do corpus de maneira ágil, sendo possível fazer login e conferir resultados no perfil do usuário, o qual salva os dados da seguinte forma:

Figura 6: Imagem de tela do relatório de resultados de corpus do TermoStat.

Corpus >> Introduction_a_l_informatique2 maribern | Aide | Déconnexion

Résultats

Liste des termes | Nuage | Statistiques | Structuration | Bigrammes

| Candidat de regroupement | Fréquence | Score (Log Odds Ratio) | Variantes orthographiques | Matrice |
|--------------------------|-----------|------------------------|--|--------------|
| rév | 1017 | 11.27 | rév | Nom |
| integer | 598 | 10.74 | integer | Nom |
| begin | 1414 | 10.5 | begin | Nom |
| else | 362 | 10.24 | else | Adjectif |
| while | 354 | 10.22 | while | Nom |
| function | 356 | 10.22 | function | Nom |
| void | 353 | 10.21 | void | Adjectif |
| begin if | 229 | 9.78 | begin if | Nom Nom |
| void | 225 | 9.76 | void | Nom |
| then | 587 | 9.62 | then | Verbe |
| procedure | 1326 | 9.59 | procedure | Nom |
| var | 568 | 9.59 | var | Nom |
| i | 174 | 9.51 | i | Adjectif |
| end | 873 | 9.51 | end | PPAdj |
| booléen | 171 | 9.49 | booléen booléens booléenne booléennes | Adjectif |
| lg | 162 | 9.44 | lg | Nom |
| boolean | 148 | 9.35 | boolean | Adjectif |
| i | 145 | 9.33 | i | Nom |
| const | 144 | 9.32 | const | Nom |
| complex | 140 | 9.29 | complex | Nom |
| gdb | 138 | 9.28 | gdb | Nom |
| property | 134 | 9.25 | property | Nom |
| def | 135 | 9.25 | def | Nom |
| programmation orientées | 131 | 9.22 | programmation orientées | Nom PPAdj |
| int | 124 | 9.17 | int | PPAdj |
| cpp | 124 | 9.17 | cpp | Nom |
| cout | 123 | 9.16 | cout | Nom |
| private | 118 | 9.12 | private | Nom |
| for i | 110 | 9.05 | for i | Nom Adjectif |

Fonte: elaboração própria.

A lista de resultados do corpus anexado no TermoStat contribuiu para uma análise de verificação de palavras-chaves e de UTEs extraídas no corpus de francês, uma vez que a exibição mais bem estruturada e mais detalhada da plataforma assegurou a compreensão e prosseguimento da análise, para então passarmos para a criação das fichas bilíngues.

A extração e análise do corpus em português foi guiada inicialmente pelos resultados que obtivemos no corpus em francês, uma vez que é a nossa língua base do glossário; contudo, vale ressaltar que ao longo da análise e do preenchimento do arquivo_controle, foram identificados resultados obtidos por meio do KWIC permitiram preencher os campos de contexto, analisar a ocorrência e encontrar UTEs relacionadas. No campo de UTEs relacionadas, adicionamos as UTEs que possuíam núcleos semelhantes (FURTADO, 2019), como por exemplo, na ficha da UTE onde “binaire” é o núcleo - “opérateur binaire”, adicionamos a seguinte ordem: “opérateur binaire - operador binário” – “arbre binaire -árvore binária”, e assim por diante. Já nos campos de definição, utilizamos a busca por definições nos próprios corpora, seguida da busca na plataforma Google Acadêmico para resultados confiáveis e fontes reconhecidas em materiais didáticos da área. Dessa forma, as definições foram adicionadas após pesquisas que nos redirecionavam para sites e plataformas de instituições de ensino técnico e universidades, validando a busca em língua de partida e para busca de equivalências entre a língua de chegada.

Após a listagem de palavras-chave e de UTEs, deu-se início ao processo de preenchimento de fichas terminológicas, seguindo o modelo apresentado na seção 3.4:

Figura 7: Visualização da ficha preenchida “variable globale-variável global” em formato .txt.

```
lJTE:
fr:
- variable globale
pt:
- variável global

UTES_relacionadas:
fr:
- variable locale
pt:
- variável local

def:
fr:
- Une variable devient globale dès qu'elle est définie en dehors de toute fonction, en général au début du fichier. Les variables globales sont utilisables par toutes les fonctions du fichier.
source:
- https://public.iutenligne.net/informatique/algorithme-et-programmation/priou/LanguageC/962_les_variables_globales.html. Acesso: 29 ago 2022.
pt:
- Uma variável é global se for definida fora de todas as funções do programa. As variáveis globais são acessíveis em qualquer ponto do programa (mesmo em outros módulos do programa).
fonte:
- https://www.lme.usp.br/~pf/algoritmos/aulas/footnotes/global-variable.html. Acesso: 29 ago 2022.

contexto:
fr:
- " Pour pouvoir affecter à une <b>variable globale</b> dans une fonction, cette variable doit faire l'ob jet d'une déclaration explicite comme variable globale de la forme global variable_globale."
pt:
- "Em geral, o uso de <b>variáveis globais</b> deve ser evitado, exceto em determinadas situações com exigências especiais de desempenho."

id:
- 38

researcher:
- Mariana Bernardo

last_update:
- 29 ago 2022
```

Fonte: elaboração própria.

As fichas foram criadas inicialmente, com o objetivo de anexar a uma plataforma de busca desenvolvida para o glossário; contudo, devido aos recursos e referenciais necessários para o desenvolvimento e manutenção de tal plataforma, foi observado que seria viável a hospedagem do glossário desenvolvida em uma ferramenta de entrada, no nosso caso, a WebS3, explicada na seção a seguir. Dessa forma, as fichas foram construídas para organização e identificação das UTEs, já salvas em formato YAML, seguindo a codificação UTF-8, permitindo a fácil edição e anexo das fichas em linguagens de programação, como apresentado em Furtado (2019).

Por fim, compilamos as fichas revisadas em formato .pdf, visando anexar na plataforma GitHub para vincular à nossa página no WebS3, apresentado na seção seguinte; contudo, as fichas originais em formato YAML permanecem disponíveis para consulta na mesma plataforma.

3.5. *LE CODE SOURCE*, O GLOSSÁRIO BILÍNGUE SOBRE PRINCÍPIOS DA PROGRAMAÇÃO

Para divulgação do glossário após exportação dos termos analisados e incorporados, foi criada uma página de acesso gratuito a partir da criação de um website simples, conectado ao banco de dados, desenvolvido na plataforma gratuita WebS3. O WebS3 disponibiliza uma gama

de opções de linguagem de programação para escolha do usuário desenvolvedor iniciante, além de cursos e tutoriais, também disponíveis gratuitamente.

O WebS3 disponibiliza uma gama de opções de templates pré-criados de acordo com o objetivo do usuário. O site também oferece vídeos e materiais sobre o estudo de linguagens de programação para escolha do usuário. No nosso projeto, nomeado “Le code source”, definimos a plataforma como meio de introdução à programação na prática, uma vez que é nossa temática principal e disponibiliza os recursos para construção de uma página de forma prática e dinâmica, como ilustrado na figura abaixo:

Figura 9: Imagem de tela do HTML básico disponibilizado pelo W3S.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="" />
6 <link rel="stylesheet" href="" />
7 <link rel="stylesheet" href="" />
8 <title>Links website</title>
9 <style>
10 body {
11     background-image: url; /* The image used for background*/
12     background-repeat: no-repeat; /* Do not repeat the image */
13     background-position: center; /* Center the image */
14     background-size: cover; /* Resize the background image to cover the entire container */
15     font-family: "Syne", sans-serif;
16 }
17
18 .container {
19     width: 100%;
20     padding-right: 15px;
21     padding-left: 15px;
22     margin-right: auto;
23     margin-left: auto;
24 }
25
26 .image-container {
27     text-align: center;
28     width: 100%;
29 }
30
31 .links-container {
32     display: flex;
33     flex-direction: column;
34     justify-content: center;
35     align-items: center;
36     gap: 20px;
37 }
38
39 .link {
40     min-width: 50% !important;
41 }
```

Fonte: elaboração própria.

O HTML (*Linguagem de Marcação de HiperTexto*) é comumente usado para desenvolver websites, tendo sido utilizado como base para nossa página. O W3S disponibiliza o código como forma de teste para os desenvolvedores aprendizes, permitindo que o código seja alterado e indicando a função de cada um dos campos (ver Fig.9) e assim que a página seja personalizada. A página foi modificada e os seguintes elementos foram adicionados:

- i. “Início” – Exibe a página principal do site;

- ii. “Acesse o glossário” – Redireciona o usuário ao GitHub, plataforma voltada para programadores, onde disponibiliza recursos de criação e colaboração de repositórios.
- iii. “Contribua com o glossário” – Permite que o usuário acesse um formulário para submissão de um novo termo, forneça referenciais terminológicos e linguísticos que validem sua tradução e, assim, colabore com a comunidade ao expandir o glossário eletrônico;
- iv. "Contato" – Fornece os dados de contato do/a pesquisador(a) responsável pelo banco de dados, caso o usuário queira solicitar mais informações.

A seguir, apresentamos informações gerais sobre o corpus e os resultados obtidos com nossas análises, para expor nossas considerações finais.

4. RESULTADOS

O processo de tradução do livro escolhido foi desafiador, contudo, de grande pertinência. A tradução do francês para o português brasileiro assegurou a colocação apropriada dos termos no glossário e nos direcionou ao início da nossa pesquisa, uma vez que a terminologia pôde ser analisada em um contexto adequado. Os capítulos selecionados do arquivo abordam as temáticas principais apresentadas nos corpora e apresentam a terminologia equivalente à trabalhada no glossário.

Com relação aos desafios da tradução, destacamos a presença dos elementos que compõem os códigos, geralmente apresentados em tabelas, para se referir à programação em linguagem C, apresentada na Unidade 4 do livro, como ilustrado na figura abaixo:

Figura 10: Imagem de tela de instrução em programação C no livro Principes de Programmation.

```
int b = 20;
int c = 0;
printf ("valeur de a dans la fonction main () = %d \n", a);
c = somme(a, b); // paramètres efectifs (a, b)
printf ("valeur de c dans main () =%d \n", c);
return 0;
}
/* Fonction pour additionner deux nombres entiers */
int somme (int a, int b) //Paramètres formels (int a, int b)
{
printf ("valeur de a dans somme() =%d \ n", a);
printf ("valeur de b dans somme() =%d \ n", b);
return a + b;
}
```

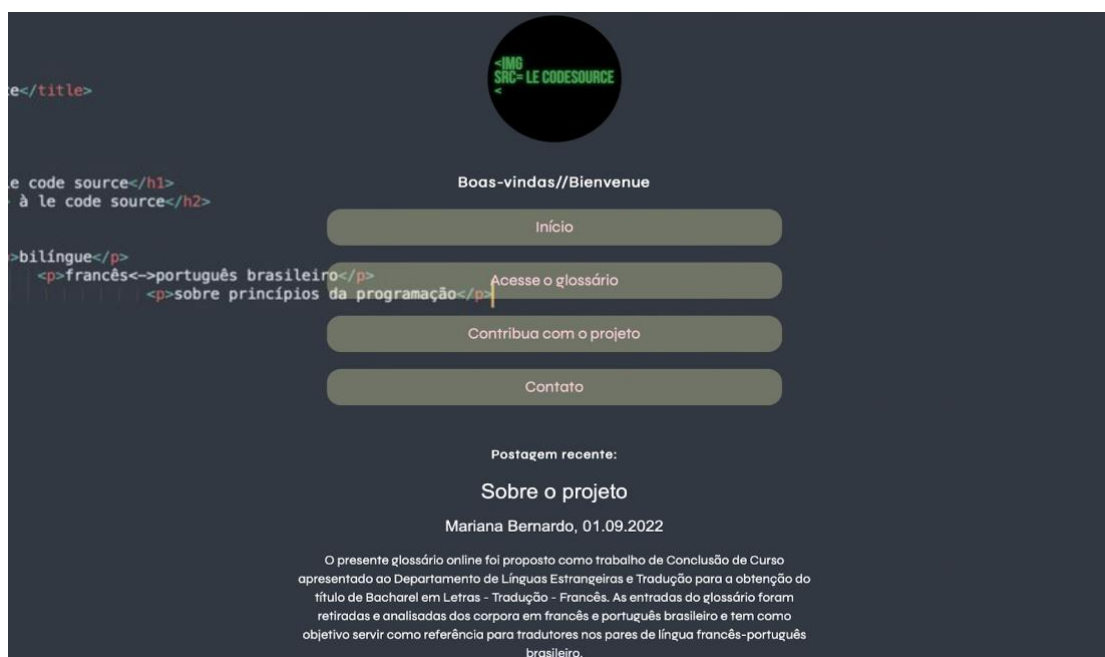
Fonte: elaboração própria.

Em programação, os códigos apresentam elementos que não são traduzíveis como os valores *char* e *int* e a função *printf* – presentes na figura ilustrada acima –, dessa forma, se fez necessária a releitura do texto para melhor compreensão e análise para tradução. Contudo, é importante ressaltar que o entendimento se fez mais acessível com os demais arquivos compilados para os corpora, uma vez que são materiais didáticos, apresentam explicações e contextos de uso tanto em francês como em português brasileiro. Dessa forma, os capítulos traduzidos foram de menos dificuldade.

Em termos gerais, a tradução teve grande sucesso com a assistência dos materiais de referência e da termbase (TB) desenvolvida com base no projeto e depois revisada e, posteriormente, transformada em um arquivo controle de fichas UTEs. É de grande importância ressaltar que o produto da tradução não apresentou inconsistências na terminologia quando processada a etapa de *QA* no Memsourc após adição da TB revisada.

Uma vez finalizada a tradução, passamos a nos concentrar totalmente em nosso glossário, fruto de pesquisas terminológicas com grande ênfase nos contextos de uso com base nos materiais didáticos selecionados para nossos corpora. Para acessar a página do *Le code source*, basta acessar o link: <https://lecodesource.w3spaces.com/index.html> e a página a seguir será exibida:

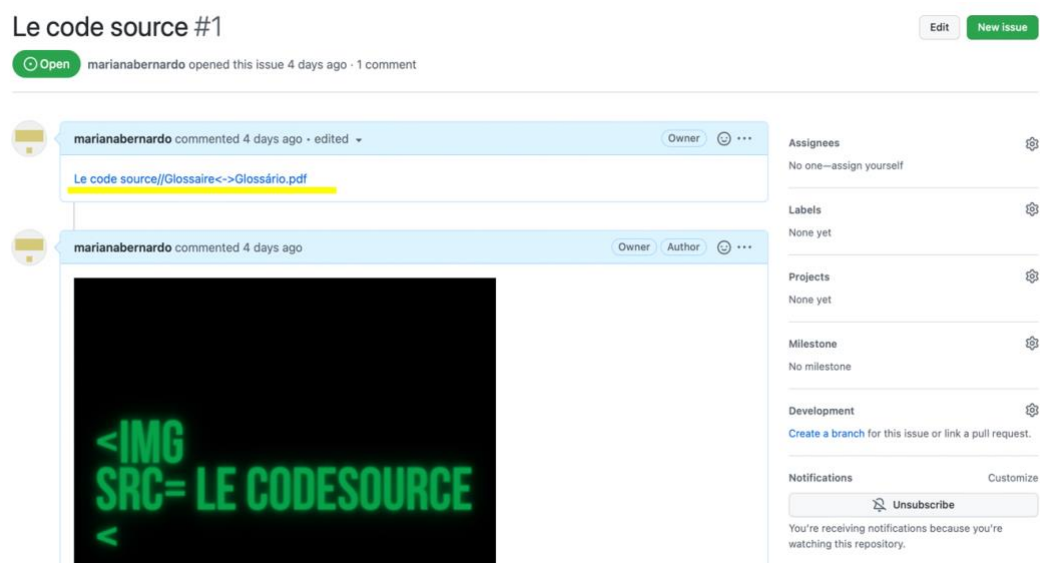
Figura 11: Página inicial do *Le code source*.



Fonte: elaboração própria.

Para acessar o glossário, basta acessar o “Acesse o glossário” no espaço indicado e assim, a página será redirecionada para a plataforma GitHub onde o arquivo .pdf está anexado e procurar a palavra ou expressão, como indicado na Figura 10. A página está e todas já estão disponíveis para consulta na página do projeto *Le code source*, no endereço: <https://github.com/marianabernardo/lecodesource/issues/1>.

Figura 12: Página inicial do *Le code source* no GitHub.



Fonte: elaboração própria.

A página redireciona ao arquivo anexado (destacado em amarelo), que automaticamente faz o download do arquivo para exibição do glossário, que inicia a amostragem em ordem alfabética, como ilustrado abaixo:

Figura 13: Página do *Le code source* em formato .pdf.

affection//atribuição

UTES relacionadas:

fr:

- instruction d'affectation

pt:

- instrução de atribuição

def:

fr:

- L'affectation donne une valeur à une variable quand'il est nécessaire de la modifier.

source:

- <http://iriaf.univ-poitiers.fr/enibook/algorithmic/learning/site/html/affection-0-index.html>. Acesso: 29 ago 2022.

pt:

- A atribuição permite dar ou alterar um valor a uma variável em um programa.

fonte:

- MACEDO, H. T. . Introdução à Ciência da Computação. São Cristóvão: Editora UFS/CESAD, 2009. v. 1. p.40.

contexto:

fr:

- " À la suite d'une telle **affection**, chaque apparition de la variable ailleurs que dans la partie gauche d'une autre affectation représente la valeur en question."

pt:

- "Os elementos de um array podem ser inicializados de três maneiras: por declaração, por **atribuição** e por entrada de valores (input)."

Fonte: elaboração própria.

A ordem das UTEs está em ordem alfabética, seguindo os campos anteriormente apresentados nas fichas na seção 3.4. No Apêndice II, mostramos uma tabela que lista integralmente as UTEs e Keywords levantadas que foram produzidas fichas neste trabalho.

4.1. TERMOS NOTÁVEIS

Como exposto nas unidades anteriores, o objetivo do corpus foi recolher materiais didáticos sobre as temáticas introdutórias em programação, aos quais estudantes tem acesso ao longo dos cursos introdutórios e apresentam fundamentos em programação e apresentação de linguagens de programação. Uma vez que é uma linguagem técnica - mesmo que introdutória - , se é esperado que siga uma terminologia consistente; contudo, durante nossas análises foram encontrados casos notáveis extraídos dos corpora e adicionados em nossa lista terminológica. Nesse sentido, apresentamos os seguintes casos:

i. computação/informática

A partir dos termos extraídos inicialmente do corpus em francês, podemos analisar em seguida o corpus em português para procura de equivalências e posterior composição das fichas terminológicas. No presente caso, *informatique*, retrata o termo que foi encontrado e definido como palavra-chave no corpus em francês e suas duas equivalências encontradas no corpus em português brasileiro:

Figura 14: Página de resultados para “informática” na ferramenta KWIC do AntConc.

| ili | Left Context | Hit | Right Context |
|-----|--|---------------|--|
| 1 | , no mínimo, adentrando o mundo da | informática, | a resposta deve ser algo do tipo: "muito!" Mas |
| 2 | , no mínimo, adentrando o mundo da | informática, | a resposta deve ser algo do tipo: Muito! Mas |
| 3 | ã Prof Jerusa Marchi Departamento de | Informática | e Estatística Universidade Federal de Santa Catarina e-mail: |
| 4 | ware livre para o desenvolvimento da | informática | e de sua opinião sobre por que as plataformas |
| 5 | balhando com os cursos Técnicos em | Informática, | na modalidade presencial e a distância, e Superior de |
| 6 | goritmo. • Entender a importância da | informática | na sociedade atual. • Apresentar os diferentes tipos de algoritmo |
| 7 | Macau) na área de Métodos Formais e | Informática | Aplicada à Saúde. Suas áreas de interesse envolvem, principalmer |
| 8 | a universitária desde 1997 na área de | informática. | Bacharel em ciência da computação pela Universidade Federal do |
| 9 | ptional em JavaJava Curso Técnico em | informática | Lógica de Programação Victorio Albani de Carvalho INSTITUTO FE |
| 10 | ima onda" que aconteceria na área da | informática. | Os idealizadores do projeto acreditavam (e acertaram) que em |
| 11 | ↳ Saída: A constante em está evolução | informática | Solução: • Digitar uma frase. • Pegar o tamanho da frase |
| 12 | a de programação : Curso Técnico em | Informática / | Victorio Albani de Carvalho. – Colatina: CEAD / Ifes, 2010. 104 p. |
| 13 | mas de várias outras ramificações da | informática, | além de ajudar em outras disciplinas do curso. A |
| 14 | Doutorado em Inteligência Artificial e | Informática | de Sinais Bio-médicos pela UFU (Universidade Federal de |
| 15 | tornou professor do departamento de | informática | do instituto Federal do Espírito Santo – Campus Colatina, onde |
| 16 | rsidade obteve o título de mestre em | Informática | em 2006. Atuou em várias empresas públicas e privadas, dentre |
| 17 | rdem alfabética. Exemplo: Entrada: A | informática | está em constante evolução Saída: A constante em está |
| 18 | ção e linguagens de programação Na | informática, | há um universo muito amplo de linguagens de programação. |
| 19 | publicação se tornou um dos livros de | informática | mais bem-sucedidos de todos os tempos. A rápida |

Fonte: elaboração própria.

Figura 15: Página de resultados para “computação” na ferramenta KWIC do AntConc.

| ili | Left Context | Hit | Right Context |
|-----|-----------------------------|------------|---|
| 1 | maticais. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 17 1.3 INTERPRETADOR Existem duas formas de |
| 2 | te: Autor LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 19 1.4 LINGUAGEM C Estudaremos neste livro |
| 3 | disciplina. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 21 1.5 AMBIENTE DE PROGRAMAÇÃO Para escrever |
| 4 | .1/2018. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 23 1.5.1 NetBeans Entre as principais características |
| 5 | p. 276.) LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 35 Na segunda linha (Linha 2) do |
| 6 | eu valor. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 37 2.1.2 Entrada e Saída Já vimos |
| 7 | lade); 7 } LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 39 Observe que as mensagens, tanto |
| 8 | acionais. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 41 Tabela 7 – Operadores Relacionais Considerando o |
| 9 | ograma. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 43 2.2 ESTRUTURAS CONDICIONAIS Estruturas condicionais, como |
| 10 | struções. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 45 2.2.3 Comando: switch O comando switch (|
| 11 | maior 7 } LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 47 2.3 COMANDOS DE REPETIÇÃO Para o |
| 12 | m geral. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 49 2.3.1 Comando: for O comando for |
| 13 | uma vez. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 51 A sintaxe do comando do- |
| 14 | de soma. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 53 1 #include <stdio.h> //define funções |
| 15 | culados. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 55 3 VETORES E MATRIZES INTRODUÇÃO Neste |
| 16 | i, temos: LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 59 1 tfindude <stdio.h> //Define funções |
| 17 | 23 } 24 } LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 61 Outro procedimento útil com vetores |
| 18 | zer isso. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 63 1 #include <stdio.h> //Define funções |
| 19 | do prinif. LICENCIATURA EM | COMPUTAÇÃO | Linguagem de programação I • 65 3.2 ACESSO AOS ELEMENTOS Vimos como |

Fonte: elaboração própria.

Como pode ser observado, ambos os termos possuem definições semelhantes, contudo, o termo "informática" apresenta uma menor ocorrência no corpus de português brasileiro (27), enquanto "computação" apresenta maior frequência (85) e o mesmo contexto e definição apresentados que o equivalente em francês; dessa forma, inicialmente foi selecionado como UTE para composição do glossário. Após revisão da lista de termos, decidiu-se criar duas fichas para o mesmo termo em francês com duas entradas diferentes em português, ao passo que

temos: “informatique-informática” e “informatique-computação”. Essa ocorrência é dada pela sinonímia na língua de partida, uma vez que o termo “informatique” se refere à área da computação e à informática em francês, mas se diverge semanticamente na língua de chegada; assim, as duas fichas foram criadas para evidenciar tal ocorrência ao usuário.

ii. *assembleur/compilateur*

Posto o caso anterior, podemos encadear os dois termos aqui apresentados como semelhantes. Após a primeira listagem de UTEs e, em seguida, com a adição de mais arquivos para compilar e atualizar os corpora, notou-se a semelhança na definição de dois termos: assembleur e compilateur.

Neste caso, em específico, fizemos uma análise contrária, o processo análise do contexto permitiu a distinção de sentidos para distinguir os termos e encontrar assim, uma solução, ou seja, nomear conceitos a partir da direção significante → significado. Dessa forma, foi observado a seguinte divergência de um para o outro: “O **compilador** analisa o código-fonte e o converte para um código-assembly (versão mnemônica da linguagem de máquina). O **montador** traduz o código assembly para código de máquina (código objeto).”

Essa diferença é válida de destaque, posto que muitas vezes as próprias definições são misturadas e, assim, acrescentadas em diversas fontes sobre a linguagem Assembly, uma linguagem também conhecida como *linguagem de montagem*.

iii. *programmation objet-orienté/orienté-objet*

Em nosso terceiro e último caso notável, apresentamos o termo “orientée-objet”, que faz referência ao conceito de *programação orientada a objetos (POO)* em português brasileiro, e inicialmente foi adicionado como “programmation objet-orientée”. Enquanto o corpus português brasileiro não apresentou ocorrências divergentes, o corpus francês gerou dois termos:

Figura 16: Página de resultados para “objet-orientée” na ferramenta KWIC do AntConc

The screenshot shows the KWIC tool interface with the search term 'objet-orientée'. The results table has 11 rows. The 'Hit' column contains the search term, and the 'Left Context' and 'Right Context' columns show the surrounding text. The 9th row is highlighted in red.

| | Left Context | Hit | Right Context |
|----|---|------------------------|--|
| 1 | ;-classe de la précédente). 7.5. - Concepts nouveaux Les langages | objet-orientés | introduisent certains concepts nouveaux reposant sur une terminologie |
| 2 | isse mère. 7.8. - L'environnement de programmation Les langages | objet-orientés | introduisent habituellement un ensemble de classes qui fournissent |
| 3 | message, classe, instance et méthode Le vocabulaire des langages | objet-orientés | fait constamment référence à cinq mots : objet, message, |
| 4 | de l'informatique chine la méthode à utiliser. Les langages | objet-orientés | ne font que généraliser et systématiser ces automatismes |
| 5 | pression est apparemment facile à comprendre. Dans les langages | objet-orientés, | c'est l'objet qui prime sur l' |
| 6 | nt automatiquement au type nouvellement créé. Dans les langages | objet-orientés, | la notion de type est étendue et contient |
| 7 | et de sous-classes qui font la force des langages | objet-orientés. | En effet, lors de la définition d'un |
| 8 | que la plupart des langages sont (sur une petite échelle) | objet-orientés. | Les opérateurs simples (+, -, *) sont généralement "objet dépendants" : la |
| 9 | pitre 3 : Les langages de l'informatique 7 .- LA PROGRAMMATION " | OBJET-ORIENTÉE" 7.1. - | Introduction La programmation OBJET-ORIENTEE est une technique |
| 10 | iques de programmation les plus modernes. Il est com-plètement | objet-orienté | et inclut, entre autres, des mécanismes permettant la |
| 11 | mais y ajoute des notions nouvelles issues de la programmation | objet-orientée (| cf §7 ci-dessous). JAVA JAVA est un langage |

Fonte: elaboração própria.

Figura 17: Página de resultados para “orientée-objet” na ferramenta KWIC do AntConc.

The screenshot shows the KWIC tool interface with the search term 'orientée-objet'. The results table has 19 rows. The 'Hit' column contains the search term, and the 'Left Context' and 'Right Context' columns show the surrounding text. The 9th row is highlighted in red.

| | Left Context | Hit | Right Context |
|----|-------------------------------------|----------------|--|
| 4 | premier est le signe Programmation | orientée objet | en C++ d'échappement '\'. Au moment de la |
| 5 | clude<iostream.h> Programmation | orientée objet | en C++ void main() { void fct(int); void |
| 6 | ane Hakim Page 17 Programmation | orientée objet | en C++ Non C=vraie ? Oui T2 T1 |
| 7 | ge 18 Exercice 2.1: Programmation | orientée objet | en C++ Ecrire un programme C++ permettant de |
| 8 | uction « continue » Programmation | orientée objet | en C++ L'instruction « continue » dans une boucle |
| 9 | ge 21 Exemple 2.4: Programmation | orientée objet | en C++ i=1; while (i<=6) { cout<<i ; i= |
| 10 | ge 23 Exemple 2.5: Programmation | orientée objet | en C++ for(i=1 ;i<=6 ;i++) { cout<<i; } |
| 11 | ième de la valeur 2. Programmation | orientée objet | en C++ Remarques: L'exécution de tout programme |
| 12 | ées-sorties binaires Programmation | orientée objet | en C++ Les fonctions d'entrées-sorties binaires |
| 13 | 9 for(i=0;i<10;i++) Programmation | orientée objet | en C++ cout<<"élément "<i<<" est : "<i<<"(p+i)<<"\ |
| 14 | ne Hakim Page 42 } Programmation | orientée objet | en C++ fclose(entree); } Exemple 2.25: Ce programme permet |
| 15 | e 44 void afficher(){ Programmation | orientée objet | en C++ cout<<"Nombre des personnes sous traitement = "<< |
| 16 | e Hakim Page 47 } } Programmation | orientée objet | en C++ Dr. Doghmane Hakim Page 48 Programmation orientée |
| 17 | iables membres ou Programmation | orientée objet | en C++ membres de données) • Fonctions appelées Méthodes (|
| 18 | .2.2. Encapsulation Programmation | orientée objet | en C++ Est le processus qui permet de |
| 19 | m Page 58 Syntaxe: Programmation | orientée objet | en C++ class nom_classe{ //déclaration des variables |

Fonte: elaboração própria.

A primeira ocorrência, listada na Figura 17, se refere ao mesmo termo, contudo, seguindo uma terminologia não oficial, ou seja, foi o termo empregado pelo autor no arquivo em específico. Pode ser observado que todas as ocorrências se referem ao mesmo arquivo no corpus do francês; dessa forma, o caso (orientée-objet) pode ser considerado uma característica do autor do material.

Além dos casos notáveis extraídos, analisados e revisados com fichas listadas acima, é importante frisar que a extração e análise do corpus em português foi guiada pelos resultados que obtivemos no corpus em francês, uma vez que tivemos cerca de 85 termos após a primeira

extração (ver Apêndice III). Decidimos levantar uma lista de UTEs extraordinárias, ou seja, UTEs a serem analisadas em futuras pesquisas, visando a expansão do corpus e promovendo a pesquisa na temática de terminologia bilingue em programação. A lista de UTEs extraordinárias foi acrescida ao arquivo controle a partir de termos observados com uma baixa ocorrência (menos que 2) ou nenhuma ocorrência no corpus, que serviu de critério para essa divisão, visto que o estudo parte de uma abordagem dirigida pela análise dos corpora, não é executável determinar traduções desses termos. Dessa forma, a listagem extraordinária parte do pressuposto que de ampliação do corpus contribuiria para a análise terminológica e para o acréscimo de entradas no glossário.

5. CONSIDERAÇÕES FINAIS

Ao longo desta apresentação de projeto, descrevemos o processo de criação de um glossário bilíngue online sobre princípios da programação, voltado para tradutores técnicos dos pares de língua francês-português brasileiro usando a Linguística de Corpus como abordagem teórico-metodológica. Dessa forma, buscamos apresentar o uso da Linguística de Corpus como fundamentação teórica principal, passando pelos fundamentos básicos da Terminologia. Utilizamos a LC pois, como exposto inicialmente em Furtado (2019), é de considerável integração com o conceito das Unidade de Tradução Especializadas e auxilia a validação dos equivalentes com base nos critérios estabelecidos por Sardinha (2004) e na compilação de textos autênticos em línguas naturais.

Com esse trabalho, partimos da tese que aponta um baixo número de materiais de consulta ou referência bilíngues voltados para tradutores nos pares de língua aqui trabalhados. Ressaltamos a necessidade da compilação e disponibilização do glossário de forma online e gratuita, promovendo assim a acessibilidade e a rápida consulta dos termos.

Para extração dos corpora compilados para o glossário, utilizamos o software gratuito de análise de corpus AntConc, que apresenta recursos que assistem o processo de análise do corpus selecionado por meio das ferramentas Wordlist, KWIC e N-Gram. Em seguida, registramos os dados analisados e obtidos em fichas terminológicas codificadas em UTF-8 em formato YAML. Para a criação do banco de dados, utilizamos a plataforma WebS3, que disponibiliza o modelo de página e o código pré-criado para construção da página de acordo com as necessidades do usuário.

Em relação aos desafios e fatores limitantes para a finalização dessas tarefas, apontamos o tempo para realização como maior desafio, uma vez que os corpora podem ser melhor analisados considerando a quantidade de dados extraídos e/ou apresentados. Além disso, para a construção das fichas descobrimos que a extração terminológica teria sido mais conveniente se o corpus fosse ainda maior, visto que precisaríamos de mais tempo para as duas tarefas, uma vez que a adição de mais arquivos compilados para os corpora já foi feita uma vez e, mesmo com tal tarefa, tivemos dificuldade em encontrar certas equivalências, o que justificou a criação da lista de UTEs extraordinárias. Outra dificuldade que tivemos foi no preenchimento do campo “definição” das fichas terminológicas, pois partimos do critério de validação a partir do uso de materiais em língua natural.

Em relação aos próximos passos, é fortemente sugerido uma nova compilação de arquivos para os corpora, para acréscimo e, assim, mais representatividade, visando incrementar a lista de UTEs e finalizar a lista de UTEs extraordinárias. Em seguida, se faz necessário a criação de mais fichas para as UTEs extraídas e analisadas. E, por fim, sugerimos o desenvolvimento de uma plataforma de busca própria para o glossário, visando assim, maior divulgação e acessibilidade e possibilitando a adição de mais informações nas fichas para consulta bilíngue.

REFERÊNCIAS

- AZENHA JR., João. **Tradução técnica e condicionantes culturais: primeiros passos para um estudo integrado**, 1999. São Paulo: Humanitas. Acesso em: 21 mai. 2022.
- BERBER, A. P. **Linguística de Corpus**. Barueri: Manole, 2004.
- BERTOLINI, Cristiano ... [et al]. **Linguagem de programação I**. 1. ed. Santa Maria, RS: UFSM, NTE, UAB, 2019.
- BYRNE, J. **Technical Translation: Usability Strategies for Translating Technical Documentation**. Holanda: Springer, 2006.
- CRUZ, C. L. S. **Estudo da terminologia das fibras e tecidos da área têxtil**. 2005. 150 p. Dissertação (Mestrado em Linguística) – Instituto de Letras, Universidade de Brasília, Brasília, 2005.
- DE MACEDO, Charles Mendes; RUELA, André Siqueira; DELGADO, Karina Valdivia. **Aplicação de Algoritmos de Agrupamento para Descoberta de Padrões de Defeito em Software JavaScript**. In: SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO (SBSI), 15, 2019, Aracajú. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 159-166.
- FISHBACK, H. Prefácio de editor convidado. Translation and Medicine, American Translators Association Series, Vol. X, pp. 1-12, 1998.
- FURTADO, A. B. D. **Glossário Multilíngue Online sobre Migração e Refúgio: uma proposta para tradutores e intérpretes**. Trabalho de Conclusão de Curso (Bacharelado em Línguas Estrangeiras Aplicadas ao Multilinguismo e à Sociedade da Informação) – Universidade de Brasília, Brasília, 2019. Disponível em: <https://bdm.unb.br/handle/10483/23240>. Acesso em 01 jun 2021.
- FURTADO, A. B. D. e GOROVITZ, S. **Primeiros Passos Para A Compilação De Um Corpus Terminológico Sobre Situações De Mobilidade: Coleta E Análise De Duas Cartilhas Informativas Multilíngues**. In: 23o Congresso de Iniciação Científica da Unb e 14o do DF, 2017, Brasília. Congresso de Iniciação Científica da Unb e Congresso de Iniciação Científica do DF, 2017.
- FURTADO, A. B. D. E TEIXEIRA, E. D. **Compilação de um Corpus Comparável Multilíngue de Migração e Refúgio**. 25º Congresso de Iniciação Científica da UNB e 16º do DF, 23 a 25 de setembro de 2019. Brasília, 2019b.
- GALEANI, L.; ASSOLINI, F. E. P. **CAT Tools e Tradução de Máquina, a revolução nas ferramentas de tradução à luz dos estudos da Linguística**. Conjecturas, [S. l.], v. 22, n. 2, p. 738–748, 2022. DOI: 10.53660/CONJ-738-B09. Disponível em: <http://conjecturas.org/index.php/edicoes/article/view/738>. Acesso em: 13 jul. 2022.
- GOMES, Gustavo Roberto da Silva. **Ampliação do Glossário Multilíngue Online Sobre Migração e Refúgio: O procedimento de criação de novas fichas**. Trabalho de Conclusão de Curso (Bacharelado em Línguas Estrangeiras Aplicadas ao Multilinguismo e à Sociedade da Informação) - Universidade de Brasília, Brasília, 2020. Disponível em: <https://bdm.unb.br/handle/10483/27167>. Acesso em: 01 jul. 2022.
- HOUNGUE, P. **Principes de Programmation**. Université Virtuelle Africaine. Bénin. 2018. p.20-52.

KORNING, K. **The dogmas of technical translation: Are they still valid?** *Hermes, Journal of Linguistics* n° 23, 1999, pp. 65-75. Disponível em: https://pure.au.dk/portal/files/9952/H23_05.pdf. Acesso em: 01 jun. 2022.

LYNNE, B; PEARSON, J. **Working with specialized language: a practical guide to using corpora**. Londres: Routledge, 2002.

Matriz Curricular de Ciência da Computação da Universidade Federal de Minas Gerais. Disponível em: <https://ufmg.br/cursos/graduacao/2377/91205/72552>. Acesso em: 01 set. 2022.

Matriz Curricular de Engenharia da Computação da Universidade Federal de Santa Maria. Disponível em: <https://www.ufsm.br/cursos/graduacao/santa-maria/engenharia-de-computacao/informacoes-do-curriculo>. Acesso em: 01 set. 2022.

Matriz Curricular de Tecnologia da Informação da Universidade Federal do Mato Grosso do Sul. Disponível em: <https://ensino.ufms.br/cursos/grade/3292>. Acesso em: 01 set. 2022.

ORMENESE, M. **TRADUTOR(A) IN MEMORIAM: USO DE MEMÓRIAS DE TRADUÇÃO**. In: XXIII Seminário de Teses em Andamento, 2018, Campinas. XXIII Seminário de Teses em Andamento, 2017. v. 8. p. 430-440.

SALGADO, A. R. **Unidades Fraseológicas Especializadas na Perspectiva da Tradução**. Tese (Mestrado em Estudos da Linguagem) – Instituto de Letras. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.

ŠANCA, F. **The Use of CAT Tools in University Translation Courses: A Case Study Based on Teaching with Memsources**. Dissertação de Mestrado - Faculdade de Artes, Charles University, Praga, 2018.

SCHNEIDER, D.; ZAMPIERI, M.; VAN GENABITH, J. **Translation Memories and the Translator: A Report on a User Survey**. *Babel*, 2019. pg. 734-762. DOI: <https://doi.org/10.1075/babel.00062.sch>.

TAGNIN, S. E. O. **A produção de dicionários direcionados pelo corpus e orientados ao tradutor como metodologia de formação de tradutores**. *Anais do X Encontro Nacional de Tradutores & IV Encontro Internacional de Tradutores (ABRAPT- UFOP, Ouro Preto, 7-10 set, 2009)*, p. 1085-1099, 2009.

TEIXEIRA, E. D. **A Linguística de Corpus a serviço do tradutor: proposta de um glossário de culinária voltado para a produção textual**. Tese (Doutorado em Estudos Linguísticos e Literários em Inglês) – Faculdade de Filosofia, Letras e Ciências Humanas, Universidade de São Paulo, São Paulo, 2008.

ZAMPIERI, Z; VELA, M. **Quantifying the Influence of MT Output in the Translators' Performance: A Case Study in Technical Translation**. *Proceedings of the EAACL 2014 Workshop on Humans and Computer-assisted Translation*. Association for Computational Linguistics. Gothenburg, Suécia. pág. 93–98. 2014.

ZARETSKAYA, A.; PASTOR, G. C., SEGHIRI, M. **Translators Requirements for Translation Technologies: Results of a User Survey**. In *Proceedings of New Horizons in Translation and Interpreting Studies*. 2015.

ZINS, C. Conceptual approaches for defining 'data', 'information', and 'knowledge'. **Journal of the American Society for Information Science and Technology**, v. 58, n. 4, p. 479-493, 2007b.

Apêndice

Apêndice I – Fichas UTEs completas do *Le code source*.

affectation//atribuição

UTEs relacionadas:

fr:

- instruction d'affectation

pt:

- instrução de atribuição

def:

fr:

- L'affectation donne une valeur à une variable quand'il est nécessaire de la modifier.

source:

- <http://iriaf.univ-poitiers.fr/enibook/algorithmic/learning/site/html/affectation-0-index.html>. Acesso: 29 ago 2022.

pt:

- A atribuição permite dar ou alterar um valor a uma variável em um programa.

fonte:

- MACEDO, H. T. . Introdução à Ciência da Computação. São Cristóvão: Editora UFS/CESAD, 2009. v. 1. p.40.

contexto:

fr:

- " À la suite d'une telle **affectation**, chaque apparition de la variable ailleurs que dans la partie gauche d'une autre affectation représente la valeur en question."

pt:

- "Os elementos de um array podem ser inicializados de três maneiras: por declaração, por **atribuição** e por entrada de valores (input)."

appel de fonction//chamada de função

UTEs relacionadas:

fr:

- fonction récursive

pt:

- chamada de função

def:

fr:

- Un appel de fonction est une expression formée à partir du nom de la fonction appelée et d'une liste d'arguments placés entre << (>> et <<) >>.

source:

- <http://cermics.enpc.fr/polys/info-96/node12.html>. Acesso: 28 ago 2022.

pt:

- Uma chamada de função contém o nome da função a ser executada seguida por uma lista de valores, chamados argumentos, os quais são atribuídos aos parâmetros da definição da função.

fonte:

- <https://panda.ime.usp.br/pensepy/static/pensepy/05-Funcoes/funcoes.html>. Acesso: 29 ago 2022.

contexto:

fr:

- "La commande step (raccourci s) a la même action que next, mais elle rentre dans les fonctions : si une instruction contient un **appel de fonction**, la commande step effectue la première instruction du corps de cette fonction."

pt:

- "Para uma função receber um array por meio de uma **chamada de função**, sua lista de parâmetros deve especificar que um array será recebido."

arbre binaire//árvore binária

def:

fr:

-Un arbre binaire est une structure composée d'ensembles de nœuds dont chacun des ensembles possède une des descriptions suivantes : ne contient aucun nœud (arbre vide); contient un nœud racine, un sous-arbre de gauche et un sous-arbre de droit.

source:

-<https://cours.etsmtl.ca/SEG/FHenri/inf145/Suppléments/arbres%20binaires.htm>. Acesso: 29 ago 2022.

pt:

- Uma árvore binária (= binary tree) é um conjunto de registros que satisfaz certas condições. Os registros serão chamados nós (poderiam também ser chamados células), contudo, as condições não serão dadas explicitamente, mas elas ficarão implicitamente claras no contexto.

fonte:

- <https://www.ime.usp.br/~pf/algoritmos/aulas/bint.html>. Acesso: 29 ago 2022.

contexto:

fr:

- "Pour les évaluations des **opérateurs binaires**, les deux opérands des opérateurs de comparaisons sont amenés à un type commun avant l'évaluation de la comparaison (flottant dans le cas d'entier et flottant)."

pt:

- "As linguagens que oferecem suporte a operadores de atribuição compostos têm versões para a maioria de seus **operadores binários**."

assembleur//montador

UTEs relacionadas:**fr:**

- compilateur

pt:

- compilador

def:**fr:**

- Programme écrit spécialement pour un ordinateur déterminé et qui permet de traduire les instructions mnémoniques en instructions numériques et aussi de les ordonner.

source:

- <https://www.cnrtl.fr/definition/assembleur>. Acesso: 30 ago 2022.

pt:

- O montador traduz o código assembly para código de máquina (código objeto). Porém, esta forma é intermediária, não podendo ser lida pelo programador, nem executada pelo computador.

fonte:

- <https://www.inf.pucrs.br/~gustavo/disciplinas/pli/material/paradigmas-aula08.pdf>. Acesso: 30 ago 2022.

contexto:**fr:**

- "Les fichiers source sont suffixés par .c, les fichiers prétraités par le préprocesseur par .i, les fichiers **assembleur** par .s, et les fichiers objet par .o."

pt:

- "Antes de serem traduzidos para linguagem de máquina pelo **montador**, é necessário que os vários módulos objetos sejam integrados de modo a formarem um único código."

base de données//banco de dados**UTEs relacionadas:****fr:**

- structure de données

pt:

- estrutura de dados

def:**fr:**

- Par définition, une base de données est un ensemble organisé d'informations, modifiables et utilisables en vue d'applications particulières, qui prend en compte les interactions logiques entre les informations issues du monde réel.

source:

-H.P. Garnir; F. Monjoie. Introduction à l'Informatique. Liège, Belgique: Université de Liège. 2006.

pt:

- Um banco de dados é uma coleção organizada de dados estruturados e normalmente armazenadas eletronicamente em um sistema de computador.

fonte:

- <https://www.oracle.com/br/database/what-is-database/>. Acesso: 28 ago 2022.

contexto:

fr:

- "Les besoins d'un utilisateur d'une **base de données** sont classiquement ceux que l'on trouve dans tout ensemble de données structurées : insertion, suppression, modification, recherche avec ou sans critère de sélection."

pt:

- "A busca por subcadeias dentro de grandes cadeias de caracteres é um problema clássico na computação, especificamente em **bancos de dados**."

booléenne//booleano

def:

fr:

- Les expressions booléennes sont les expressions qui ont pour valeur l'un des deux booléens true (vrai) ou false (faux).

source:

- <https://www.fil.univ-lille.fr/~wegrzyno/portail/InitProg/Doc/Poly/poly.pdf>. Acesso: 29 ago 2022.

pt:

- Variáveis do tipo boolean podem assumir os valores true ou false. O valor default para um atributo booleano de uma classe, se não especificado, é false.

fonte:

- <https://www.dca.fee.unicamp.br/cursos/PooJava/sintaxe/boolean.html>. Acesso: 29 ago 2022.

contexto:

fr:

- "Pour calculer la fonction f à partir d'un schéma de circuits logiques, il suffit d'indiquer à la sortie de chaque opérateur (circuit de base) la valeur de l'expression **booléenne** en cours."

pt:

- "Lógicos - são também chamados de dados **booleanos** e podem assumir os valores verdadeiro ou falso."

boucle//laço

UTES relacionadas:

fr:

- instruction de boucle

pt:

- instrução do laço

def:

fr:

- Dans un algorithme, utiliser une boucle permet de recommencer plusieurs fois un bloc d'instructions.

source:

- <https://ww2.ac-poitiers.fr/math/spip.php?article444>. Acesso em: 29 ago 2022.

pt:

- Laços (loops ou comandos de repetição) são estruturas de repetição que permitem executar o mesmo comando repetidas vezes.

fonte:

- <https://www.inf.pucrs.br/~pinho/Laprol/ComandosDeRepeticao/Repeticao.html>. Acesso em: 29 ago 2022.

contexto:

fr:

- "Il y a deux instructions de saut qui sont importantes dans la programmation. Ce sont: les instructions continue et break. Break arrête la **boucle** tandis que continue sort de la boucle et recommence à l'exécuter à l'itération suivante."

pt:

- "No while o teste é feito antes da execução do **laço**; já no do...while primeiro executa-se o laço e só depois o teste é feito para verificar se o laço continuará a ser executado."

cadeia de caractere//cadeia de caractere

def:

fr:

- Les chaînes de caractères sont des séquences finies de caractères. Le nombre de caractères composant une chaîne de caractères est la longueur de cette chaîne.

source:

- <https://www.fil.univ-lille.fr/~wegrzyno/portail/InitProg/Doc/Poly/poly.pdf>. Acesso em: 28 ago 2022.

pt:

- Uma cadeia de caracteres (também conhecida por string), é definida por uma sequência de letras e símbolos com aspas no programa.

fonte:

- <https://www.ic.unicamp.br/~ducatte/mc102/aula18.pdf>. Acesso em: 28 ago 2022.

contexto:

fr:

- "Comme pour les tuples, on peut concaténer deux **chaînes de caractères** à l'aide de + pour en produire une troisième, la longueur de la chaîne est donnée par len, et l'accès aux caractères est similaire."

pt:

- "As variáveis que armazenam as **cadeias de caracteres** podem ser inicializadas automaticamente pelo programa ou podem receber um valor por meio do teclado."

code machine//código de máquina

UTEs relacionadas:

fr:

- code source

pt:

- código fonte

def:

fr:

- Le code machine c'est le langage natif du processeur. Il est aussi appelé langage machine.

source:

<https://www.techno-science.net/definition/5326.html>. Acesso: 29 ago 2022.

pt:

- O código de máquina consiste em uma sequência de bytes que correspondem a instruções que serão executadas pelo processador.

fonte:

<https://insper.github.io/Z01.1/Teoria-Linguagem-de-Maquina/>. Acesso: 29 ago 2022.

contexto:

fr:

- "Une coclasse est une classe chargée d'implémenter une ou plusieurs interfaces. C'est elle qui va "contenir" le **code machine** et donc constituer le moule de fabrication de l'objet COM."

pt:

- "O código fonte é traduzido para **código de máquina**, que é executado pelo processador"

code source//código fonte

UTEs relacionadas:

fr:

- code machine

pt:

- código de máquina

def:

fr:

- Ensemble d'instructions écrit dans un langage de programmation qui devient un code objet qui peut être exécuté par l'ordinateur.

source:

- <https://www.lalanguefrancaise.com/dictionnaire/definition/code-source>. Acesso: 29 ago 2022.

pt:

- O código fonte é criado em um editor de textos e contém os comandos da linguagem de programação, servindo como entrada para o compilador do programa.

fonte:

- <https://www.inf.pucrs.br/~pinho/Laprol/ConceitosBasicos/ConceitosBasicos.htm>. Acesso: 29 ago 2022.

contexto:

fr:

- "Les effets d'une modification du **code source** ou d'une erreur durant l'exécution dans un module peuvent se propager à un nombre plus ou moins important de modules en suivant le graphe de liaison."

pt:

- "O compilador analisa o **código-fonte** e converte-o para um código-objeto (uma versão em linguagem de máquina apropriada para o computador)."

compilateur//compilador

UTES relacionadas:

fr:

- assembleur

pt:

- montador

def:

fr:

- Les compilateurs sont des programmes qui transforment du code source en code exécutable par une machine.

source:

- <https://info.uqam.ca/~privat/INF5000/01-intro.pdf>. Acesso: 30 ago 2022.

pt:

- Compiladores produzem a partir do programa de entrada, outro programa que é equivalente ao original, porém numa linguagem que é executável. Este programa resultante pode ser em uma linguagem que é diretamente executável, tal como linguagem de máquina, ou indiretamente executável, tal como outra linguagem para a qual já existe um tradutor.

fonte:

- <https://www.inf.pucrs.br/~gustavo/disciplinas/pli/material/paradigmas-aula08.pdf>. Acesso: 30 ago 2022.

contexto:

fr:

- "Lorsqu'une méthode à liaison statique est invoquée dans le corps d'un programme, le **compilateur** établit immédiatement dans le code appelant l'adresse précise et connue du code de la méthode à invoquer."

pt:

- "O especificador de classe de armazenamento é uma recomendação ao **compilador** para armazenar uma variável em um dos registros do computador."

débogueur//depurador

def:

fr:

-Un débogueur est un outil de développement spécialisé qui permet d'inspecter un code et corriger les erreurs.

source:

<https://www.larousse.fr/dictionnaires/francais/débogueur/21835>. Acesso: 29 ago 2022.

pt:

- O depurador elimina erros em um programa. Apesar das próximas seções estarem organizadas de acordo com o tipo do erro, alguma técnicas são aplicáveis em mai do que uma situação.

fonte:

https://panda.ime.usp.br/pensepy/static/pensepy/Appendices/app_a.html. Acesso: 29 ago 2022.

contexto:

fr:

- "Pour utiliser le **débogueur**, il est nécessaire de modifier les options du compilateur afin d'inclure des informations de débogage dans les fichiers objet et exécutables (Outils> Options> Compiler compilateur)."

pt:

- "A compilação condicional é usada normalmente como uma ajuda na depuração (debugging) do programa. Muitas implementações do C fornecem **depuradores** (debuggers)."

déclaration de variable//declaração de variável

def:

fr:

-La déclaration d'une variable sert à définir son type et par conséquent la manière dont elle sera représentée en mémoire.

source:

- <http://cours.thirion.free.fr/Cours/Premieres-Notions/LZ-Notion-De-Variable.html>. Acesso: 30 ago 2022.

pt:

-Variáveis representam uma forma de identificar por um nome simbólico uma região da memória que armazena um valor sendo utilizado por uma função.

fonte:

- <https://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node124.html>. Acesso: 29 ago 2022.

contexto:

fr:

- "Contrairement à d'autres langages tels que C, Ada ou Pascal, en Caml il n'est pas nécessaire d'apprendre de nouvelle forme syntaxique pour déclarer une nouvelle fonction. On utilise la même forme let que pour la **déclaration des variables**."

pt:

- "As variáveis globais são criadas colocando **declarações de variáveis** fora de qualquer definição de função, e conservam seus valores ao longo de toda a execução do programa."

déclaration//declaração

UTEs relacionadas:

fr:

- déclaration de variable

pt:

- declaração de variável

def:

fr:

- Type d'instruction particulière d'un langage synthétique permettant de préciser le type des variables, leur degré de permanence, la structure d'un fichier, etc.

source:

- [https://www.cnrtl.fr/definition/déclaration](https://www.cnrtl.fr/definition/d%C3%A9claration). Acesso: 26 ago 2022.

pt:

- Declarações são mensagens para o compilador que informam os atributos das variáveis e requerem espaço para tais mensagens.

fonte:

- C++: como programar. H.M. Deitel; P.J. Deitel. 3.ed.Porto Alegre: Bookman, 2001.

contexto:

fr:

- "Une **déclaration** d'un nouveau type de données sert à associer un identificateur à un type de données construit par l'utilisateur."

pt:

- "Os elementos de um array também podem ser inicializados na **declaração** do array com um sinal de igual e uma lista de inicializadores separada por vírgulas (entre chaves)."

erreur de compilation//erro de compilação

def:

fr:

- Une erreur de compilation se produit lorsqu'on tente de modifier la valeur d'une constante qui est déjà initialisée, ou quand'on tente d'utiliser une variable qui n'a pas encore été initialisée.

source:

- <https://www.ukonline.be/cours/java/apprendre-java/chapitre1-4>. Acesso: 26 ago 2022.

pt:

- Os erros de compilação ou erros de sintaxe são transgressões às regras definidas da linguagem de programação. É causado quando o compilador não reconhece uma instrução e também é apontado pelo compilador para correção na instrução.

fonte:

- C++: como programar. H.M. Deitel; P.J. Deitel. 3.ed.Porto Alegre: Bookman, 2001.

contexto:

fr:

- "Il existe trois types d'erreurs de base à savoir : syntaxe ou **erreurs de compilation**, erreurs d'exception à l'exécution et erreurs logiques. Un programme doit être testé pour tous ces types d'erreurs."

pt:

- "Por escopo, pode-se entender as partes de um programa em que uma constante e também variáveis são compreendidas e podem ser utilizadas, sem acarretar **erros de compilação**."

exécuter des tâches//executar tarefas

UTEs relacionadas:

fr:

- exécuter un programme
- exécuter une instruction

pt:

- executar um programa
- executar uma instrução

def:

fr:

- À l'intérieur d'un programme, partie représentant un ensemble logique de fonctions, pouvant s'exécuter le cas échéant en même temps que d'autres tâches du même programme.

source:

- <https://www.cnrtl.fr/definition/t%C3%A0che>. Acesso: 31 ago 2022.

pt:

- Uma tarefa é definida como sendo a execução de um fluxo sequencial de instruções, construído para atender uma finalidade específica: realizar um cálculo complexo, a edição de um gráfico, a formatação de um disco, etc..

fonte:

- <http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-04.pdf>. Acesso: 31 ago 2022.

contexto:

fr:

- "Dans un OS interactif comme Windows, Linux, Mac OS par exemple, la préemption est

fondamentale car il y a beaucoup d'intervention de l'utilisateur pendant l'**exécution des tâches.**"

pt:

- "Na programação estruturada, a divisão de trabalho é um processo chamado de modularização. Nesse processo, divide-se o programa em partes ou módulos que **executam tarefas** específicas."

exécuter un programme//executar um programa

UTES relacionadas:

fr:

- exécuter des tâches
- exécuter une instruction

pt:

- executar tarefas
- executar uma instrução

def:

fr:

- Le processeur charge le programme en mémoire pour ensuite commencer à l'exécuter. En plus de charger le programme, il va charger également en mémoire des données dont il a besoin, donc une fois qu'il a tout ça dans sa mémoire, le processeur va commencer à travailler.

source:

- <https://lms.fun-mooc.fr/c4x/inria/41014/asset/C021TV-I4-S2.pdf>. Acesso em: 28 ago 2022.

pt:

- Um programa é uma sequência de instruções, é algo passivo dentro do sistema. Ele não altera o seu próprio estado. O processo é um elemento ativo. O processo altera o seu estado à medida que executa um programa. é o processo que faz chamadas de sistema, ao executar os programas.

<https://www.inf.pucrs.br/flash/lapro2/trab021/trabalho.html>. Acesso em: 28 ago 2022.

contexto:

fr:

- "Dans le cas d'un langage compilé, c'est le micro-processeur qui **exécute le programme** et on ne peut pas l'arrêter à chaque instruction !"

pt:

- "No processamento de lotes de usuário único (single-user batch processing), o computador **executa um programa** simples de cada vez enquanto processa os dados em grupos ou lotes (batches)."

fichier binaire//arquivo binário

UTEs relacionadas:**fr:**

- fichier d'en-tête

pt:

- arquivo de cabeçalho

def:**fr:**

-Fichier contenant des données codées sous forme de séquence de bits (zéros et uns) et non du texte brut. Un fichier binaire, tel qu'une DLL ou un fichier exécutable, contient des informations pouvant être directement chargées ou exécutées par un ordinateur.

source:

- <https://support.esri.com/fr/other-resources/gis-dictionary/term/b509a9e3-f664-4c6b-8f74-638f0f4ac07e>. Acesso em: 29 ago 2022.

pt:

- Arquivos binários são usados para armazenar informações mais complexas como um código executável e são definidos por não conter informações textuais.

fonte:

- http://wiki.inf.ufpr.br/maziero/doku.php?id=prog2:arquivos_binarios. Acesso em: 29 ago 2022.

contexto:**fr:**

- "FTP (File Transfert Protocol) est un protocole permettant de rapatrier sur votre machine ou d'expédier à partir de votre machine des **fichiers binaires** quelconques."

pt:

- "**Arquivos binários** devem ser lidos por programas especiais, que convertem a cadeia de bits em informações compreensíveis."

fichier d'en-tête//arquivo de cabeçalho**UTEs relacionadas:****fr:**

- fichier binaire

- fichier exécutable

pt:

- arquivo binário

- arquivo executável

def:**fr:**

- Un fichier d'entête contient les déclarations des fonctions et variables qui doivent être utilisées à l'extérieur du fichier .c ou elles sont définies.

source:

- <https://dept-info.labri.fr/~baudon/InitProg/2004/fichiers.html>. Acesso em: 30 ago 2022.

pt:

- O arquivo de cabeçalho são aqueles que o compilador inclui no início do código e são caracterizados pelo .h ao final.

fonte:

- <http://mtm.ufsc.br/~azeredo/cursoC/aulas/c750.html>. Acesso em: 30 ago 2022.

contexto:

fr:

- "En pratique, le **fichier d'en-tête** ne contient pas seulement les déclarations des fonctions mais aussi les définitions des nouveaux types (comme les structures) utilisés par le fichier séparé."

pt:

- "Cada biblioteca padrão tem um **arquivo de cabeçalho** correspondente contendo os protótipos de todas as funções daquela biblioteca e definições dos vários tipos de dados e constantes necessários por elas."

fonction réursive//função recursiva

def:

fr:

- Une fonction réursive est essentiellement une fonction qui s'appelle elle-même, ainsi les appels successifs à f s'empile dans la pile d'appels.

source:

-SVARTZ. J. Cours d Informatique pour Tous. Lycée Masséna: Nice, França. Pg. 157.

pt:

- Uma função é dita recursiva quando dentro dela é feita uma ou mais chamadas a ela mesma.

fonte:

- http://www.vision.ime.usp.br/~pmiranda/mac2166_1s17/aulas/P3/aulas_P3.html. Acesso: 31 ago 2022.

contexto:

fr:

- "Dans le cas d'une **fonction réursive**, on essaie donc s'il est nécessaire d'écrire une version dérursive (ou itérative) de la fonction."

pt:

- "Para uma recursão terminar, cada vez que uma **função recursiva** chamar a si mesma com uma versão ligeiramente mais simples do problema original, a sequência de problemas cada vez menores deve convergir para o caso básico."

fonction//função

UTEs relacionadas:

fr:

- fonction réursive

- appel de fonction

pt:

- função recursiva
- chamada de função

def:

fr:

- Une fonction est une partie d'un programme pouvant prendre en entrée des variables (appelées, dans ce contexte, arguments de fonction) et retournant parfois une valeur.

source:

- [https://www.editions-
eni.fr/open/mediabook.aspx?idR=34ab37822c6d52ca6d641225c4a0ae3e](https://www.editions-eni.fr/open/mediabook.aspx?idR=34ab37822c6d52ca6d641225c4a0ae3e). Acesso: 31
ago 2022.

pt:

- Funções podem ser descritas como pequenos algoritmos (ou procedimentos) dentro de um programa.

fonte:

- BERTOLINI, Cristiano ... [et al]. Linguagem de programação I. 1. ed. Santa Maria, RS: UFSM, NTE, UAB, 2019.

contexto:

fr:

- "Comme un identifiant, un nom de **fonction** est le nom effectif de la fonction qui est utilisé par d'autres programmes chaque fois qu'ils ont besoin de ses services."

pt:

- "Os rótulos podem ser usados em qualquer lugar de uma função na qual aparecem, mas não pode ser feita qualquer referência a eles fora do corpo da função."

informatique//computação

UTEs relacionadas:

fr:

- informatique

pt:

- informática

def:

fr:

- Science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines technique, économique et social.

source:

- <https://www.cnrtl.fr/definition/informatique>. Acesso: 26 ago 2022.

pt:

- Computação é a ciência que estuda e implementa os algoritmos e as suas aplicações nos mais diversos ramos do conhecimento humano e também em máquinas.

fonte:

- <https://www.gta.ufrj.br/~pisa/index.php/computacao>. Acesso: 26 ago 2022.

contexto:

fr:

- "On admet généralement que l'ère de l'**informatique** qui couvre peu de décennies se divise en plusieurs générations essentiellement marquées par des avancées technologiques"

pt:

- "A tecnologia do chip de silício tornou a **computação** tão econômica que aproximadamente 150 milhões de computadores de uso geral estão sendo empregados em todo o mundo, ajudando as pessoas no comércio, indústria, governo e em suas vidas particulares."

instruction d'affectation//instrução de atribuição

UTES_relacionadas:

fr:

- instruction de débogage
- instruction d'contrôle
- opérateur d'affectation

pt:

- instrução de declaração
- instrução de controle
- operador de atribuição

def:

fr:

- L'instruction d'affectation simple utilise l'opérateur d'affectation simple qui est le '=' en C. C'est ce que l'on appelle un opérateur binaire car il comporte à sa gauche un premier opérande et à sa droite un deuxième opérande.

source:

http://ressources.unit.eu/cours/Cfacile/co/ch4_p1_2.html. Acesso: 29 ago 2022.

pt:

- O comando de atribuição é utilizado para armazenar um valor em uma variável.

fonte:

https://olimpiada.ic.unicamp.br/saci/cursos/intro_js/4/. Acesso: 29 ago 2022.

contexto:

fr:

- "Attention à ne pas confondre le symbole := utilisé en Caml pour l'**instruction d'affectation**, avec le symbole = utilisé comme opérateur de comparaison, ainsi que dans la déclaration de variables."

pt:

- "A expressão de inicialização é uma **instrução de atribuição** executada apenas uma vez, no início do laço."

instruction en langage machine// instrução em linguagem de máquina

UTEs relacionadas:

fr:

- code machine

pt:

- código de máquina

def:

fr:

- Les instructions en langage source sont compilées ou regroupées dans le code machine stocké sous la forme d'un fichier exécutable jusqu'à ce qu'un utilisateur lance l'instruction d'exécution via le système d'exploitation (OS). A chaque instruction de code machine, l'unité centrale (l'UC) exécute une opération simple.

source:

<https://www.lemagit.fr/definition/code-machine-langage-machine>. Acesso: 29 ago 2022.

pt:

- Na linguagem de máquina, a representação dos dados e das operações (instruções) que constituem um programa, é baseada no sistema binário, que é a forma compreendida e executada pelo hardware do sistema.

fonte:

<http://professores.dcc.ufla.br/~monserrat/icc/Capitulo3.html>. Acesso: 29 ago 2022.

contexto:

fr:

- "L'assembleur est utilisé pour convertir les instructions en **langage machine**."

pt:

- "A **instrução em linguagem de máquina** verifica o comprimento e imprime a string traduzindo cada número de dois dígitos em seu caractere equivalente."

instruction//instrução

UTEs relacionadas:

fr:

- instruction en langage machine

- instruction d'affectation

pt:

- instrução em linguagem de máquina

- instrução de atribuição

def:

fr:

- Suite de caractères écrite dans un code appelé langage de programmation, qui, dans un système d'exploitation ou un logiciel d'application, permet d'exécuter une fonction particulière.

source:

- https://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=8384350. Acesso: 26 ago 2022.

pt:

- A operação de uma CPU é determinada pelas instruções que ela executa, conhecidas como instruções de máquina ou instruções do computador.

fonte:

- http://www.univasf.edu.br/~fabio.nelson/arq/aoc1/aula_10.pdf. Acesso: 26 ago 2022.

contexto:

fr:

- "La programmation peut être définie comme l'action d'utiliser un langage de programmation pour écrire des **instructions** qui peuvent être exécutées par l'ordinateur dans le but de résoudre des problèmes en appliquant correctement la syntaxe et la sémantique du langage."

pt:

- "Um interpretador é um programa que lê uma **instrução** de outro programa em linguagem de alto nível, determina a operação a ser realizada pela instrução e executa a operação imediatamente."

langage de bas niveau//linguagem de baixo nível

UTEs relacionadas:

fr:

- langage de haut niveau

pt:

- linguagem de alto nível

def:

fr:

- La langage de bas niveau sert à formuler des instructions directement interprétables par l'ordinateur et dont la conception est fondée sur la structure du processeur utilisé par un type particulier d'ordinateur.

source:

- https://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=8370858. Acesso: 26 ago 2022.

pt:

- São denominadas de linguagens de baixo nível as linguagens com mais semelhança e proximidade com representação binária (linguagem de máquina), como a linguagem Assembly.

fonte:

- SOUSA, Bruno Jefferson de; JÚNIOR, José Jorge Lima Dias; FORMIGA, Andrei de Araújo. Introdução à Programação. Editora da Universidade Federal da Paraíba: João Pessoa, 2014.

contexto:

fr:

- "La contrepartie dans l'utilisation d'un **langage de bas niveau** est la longueur du code qui est en général bien plus importante."

pt:

- "Os primeiros computadores foram criados a partir de 1940 onde eram programados na linguagem Assembly, também chamada de linguagem de montagem ou **linguagem de baixo nível**."

langage de haut niveau//linguagem de alto nível

UTEs relacionadas:

fr:

- langage de bas niveau

pt:

- linguagem de baixo nível

def:

fr:

- Langage de programmation indépendant du type d'ordinateur utilisé et dont la syntaxe, s'approchant de celle du langage naturel, facilite l'écriture et la compréhension de programmes.

source:

- https://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=2071750. Acesso: 26 ago 2022.

pt:

- São denominadas de linguagens de baixo nível as linguagens mais próximas às linguagens naturais, compostas de um conjunto de palavras-chave e símbolos que estabelecem os comandos e instruções na construção de programas.

fonte:

- SOUSA, Bruno Jefferson de; JÚNIOR, José Jorge Lima Dias; FORMIGA, Andrei de Araújo. Introdução à Programação. Editora da Universidade Federal da Paraíba: João Pessoa, 2014.

contexto:

fr:

- "Les **langages de haut niveau** ont différents attributs ou caractéristiques qui comprennent : la portabilité, la lisibilité, le support de la concurrence, le support du mixage de langage, la fiabilité, la modularité, le support du temps réel et l'orthogonalité."

pt:

- "As **linguagens de alto nível** permitem aos programadores escrever instruções que se parecem com o idioma inglês comum e contêm as notações matemáticas normalmente usadas."

mnémonique//mnemônico

def:**fr:**

- C'est une instruction en langage assembleur, facilement compréhensible et mémorisable (symbole court d'une à quelques lettres), convertie en code binaire au moment de l'assemblage.

source:

- <https://www.lalanguefrancaise.com/dictionnaire/definition/mnemonic>. Acesso: 26 ago 2022.

pt:

- Na linguagem Assembly se faz uso de palavras abreviadas, nomeadas mnemônicos, indicando a operação a ser realizada pelo processador.

fonte:

- <http://professores.dcc.ufla.br/~monserrat/icc/Capitulo3.html>. Acesso: 26 ago 2022.

contexto:**fr:**

- "Les améliorations sur cette catégorie de langages se sont limitées à construire des langages symboliques (langage avec **mnémorique**) et des macro-assembleurs."

pt:

- "Normalmente, os nomes dos procedimentos são palavras ou pequenas frases que procuram associá-los de forma <**mnemônica** à tarefa realizada."

opérateur d'affectation//operador de atribuição**UTEs relacionadas:****fr:**

- instruction d'affectation
- instruction d'contrôle

pt:

- instrução de declaração
- instrução de controle

def:**fr:**

- L'affectation =, et ses dérivées +=, *=, etc., sont des opérateurs binaires. L'affectation est prédéfinie pour toutes les classes, et représente alors une copie terme à terme des membres de la classe.

source:

- <https://www.ljll.math.upmc.fr/cours/cpp/Sections/Sect07-B8.html>. Acesso: 29 ago 2022.

pt:

- Um operador de atribuição atribui um valor ao operando à sua esquerda baseado no valor do operando à direita.

fonte:

- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Expressions_and_Operators#operador_atribuicao. Acesso: 29 ago 2022.

contexto:

fr:

- "Les constantes peuvent être définies en utilisant un **opérateur d'affectation** (=), ou un mot clé const ou la directive #define."

pt:

- "A linguagem C fornece vários **operadores de atribuição** que ajudam a abreviar determinados tipos comuns de expressões aritméticas de atribuição."

opérateur d'incrémentation//operador de incremento

UTEs relacionadas:

fr:

- opérateur d'affectation

pt:

- operador de atribuição

def:

fr:

-L'opérateur d'incrémentation (++) est un opérateur unaire qui permet d'ajouter 1 au contenu de l'opérande.

source:

https://perso.univ-lyon1.fr/marc.buffat/COURS/C_HTML/node25.html. Acesso: 29 ago 2022.

pt:

-São operadores adicionais (++) que podem ser usados para encurtar as operações e podem ser colocados depois do nome da variável.

fonte:

https://www.inf.ufpr.br/ci208/NotasAula/notas-15_Operadores_Expressoes_Es.html. Acesso: 29 ago 2022.

contexto:

fr:

- "Les **opérateurs d'incrémentation** ++ et de décrémentation -- s'utilisent aussi bien en suffixe (i++) qu'en préfixe (++i)."

pt:

- "A linguagem C fornece o **operador de incremento**, ++, e o operador de decremento, -, para incrementar ou decrementar uma variável de 1."

opérateur//operador

UTEs relacionadas:

fr:

- opérateur binaire
- opérateur d'incrément
- opérateur d'affectation

pt:

- operador binário
- operador de incremento
- operador d'affectation

def:

fr:

- L'opérateur peut effectuer des opérations arithmétiques, booléennes ou agir sur des chaînes de caractères. Les caractères de leurs expressions peuvent être des symboles ou de ponctuation.

source:

- <https://fr-academic.com/dic.nsf/frwiki/1264912>. Acesso: 26 ago 2022.

pt:

- Um operador, em termos gerais, simboliza uma operação efetuada sobre variáveis e constantes, executando cálculos e manipulação sobre os dados.

fonte:

- <https://docplayer.com.br/190113657-Logica-de-programacao-versao-2.html>. Acesso: 30 ago 2022.

contexto:

fr:

- "La surcharge des **opérateurs** en C++ permet de donner aux opérateurs une signification particulière pour un type de données."

pt:

- "O operador de ponteiro de estrutura e o operador de membro de estrutura, juntamente com os parênteses e colchetes ([]) usados para subscritos de arrays, são **operadores** que possuem a maior precedência e fazem associações da esquerda para a direita."

programmation orientée objet//programação orientada a objetos

UTEs relacionadas:

fr:

- programmation structurée

pt:

- programação estruturada

def:

fr:

- La programmation orientée objet, souvent abrégée POO, permet de concevoir une application sous la forme d'un ensemble de briques logicielles appelées des objets qui jouent un rôle précis et peut communiquer avec les autres objets.

source:

- https://www.univ-usto.dz/images/coursenligne/poo_hy.pdf. Acesso: 30 ago 2022.

pt:

- Um programa orientado a objetos é composto por um conjunto de objetos que interagem através de “trocas de mensagens”. Na prática, essa troca de mensagem traduz-se na aplicação de métodos a objetos.

fonte:

- <https://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>. Acesso: 29 ago 2022.

contexto:

fr:

- "Le polymorphisme est le troisième aspect essentiel d'un langage de **programmation orientée objet**, après l'abstraction des données et l'héritage."

pt:

- "Já a partir da década de 80 as linguagens se consolidaram e também começou a se popularizar outros paradigmas de programação, como a **programação orientada a objetos** que consiste na composição e interação entre unidades de código chamadas objeto."

programmation structurée//programação estruturada

UTEs relacionadas:

fr:

- programmation orientée objet

pt:

- programação orientada a objetos

def:

fr:

- La programmation structurée est un style de programmation dans lequel la structure du programme est construite de la façon la plus claire que possible.

source:

- https://www.mcours.net/cours/pdf/info/Programmation_Structuree.pdf. Acesso: 28 ago 2022.

pt:

- A programação estruturada estabelece uma disciplina de desenvolvimento de algoritmos que facilita a compreensão de programas através do número restrito de mecanismos de controle da execução de programas.

fonte:

- <https://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node7.html>. Acesso: 28 ago 2022.

contexto:

fr:

- "Avec

l'avènement des ces langages tels que le C, la **programmation structurée** est devenue très populaire et a été la principale technique des années 1980. "

pt:

- "Com a evolução dos computadores, de seus sistemas operacionais e da complexidade, surgiram as linguagens estruturadas, ou **programação estruturada**, que permitiram o desenvolvimento de sistemas mais interativos, com maior organização e funcionalidade."

récursive//recursivo

UTEs relacionadas:

fr:

- fonction récursive

pt:

- função recursiva

def:

fr:

- La récursive est toute fonction ou procédure qui s'appelle elle même.

source:

- <https://www.lri.fr/~hivert/COURS/CFA-L3/02-Recursivite.pdf>. Acesso: 28 ago 2022.

pt:

- A recursiva é uma função que é definida em termos de si mesma. A Recursividade por sua vez, é o mecanismo básico para repetições nas linguagens funcionais.

fonte:

- <http://www.decom.ufop.br/romildo/2012-1/bcc222/slides/06-recursividade.pdf>. Acesso: 28 ago 2022.

contexto:

fr:

- "L'usage de la récursivité présente deux inconvénients : il faut faire attention à ce que les appels **récurifs** ne se chevauchent pas, et prendre garde à ne pas faire un trop grand nombre d'appels récurifs imbriqués."

pt:

- "Normalmente, um método **recursivo** é escolhido em detrimento de um método iterativo quando reflete melhor o problema e resulta em um programa mais fácil de entender e depurar."

obs:

- Foram identificadas também no corpus as variações verbais, adverbiais e nominais da UTE como: *récursivité; récurion; récurivement*.

séquence d'échappement//sequência de escape

UTEs relacionadas:

fr:

- séquence d'instruction

pt:

- sequência de instrução

def:

fr:

- Les caractères de séquence d'échappement sont des caractères non imprimables qui spécifient une autre interprétation du caractère suivant comme séquence de caractères d'échappement.

source:

- <https://www.delftstack.com/fr/howto/mysql/escape-sequences-in-mysql/#d%25C3%25A9finition-de-la-s%25C3%25A9quence-d%25C3%25A9chappement>. Acesso: 29 ago 2022.

pt:

- Sequências de escape “\n” são representada por um caractere de nova linha o “n”, fazendo que o cursor de saída da tela mova-se para o começo de uma nova linha.

fonte:

- <https://www.devmedia.com.br/system-out-objeto-de-saida-em-java/25240>. Acesso: 29 ago 2022.

contexto:

fr:

- "Il y a beaucoup d'autres caractères non-imprimables et C représente ces caractères en utilisant une **séquence d'échappement** (\)."

pt:

- "Uma instrução printf com várias **sequências de escape** \n pode imprimir várias linhas."

structure de contrôle//estrutura de controle

UTEs relacionadas:

fr:

- structure de sélection

pt:

- estrutura de seleção

def:

fr:

- La structure de contrôle est la seule instruction qui a des effets sur le contenu de la mémoire est l'affectation. On distingue les structures de contrôle conditionnelles et itératives.

source:

- <http://deptinfo.cnam.fr/Enseignement/CycleProbatoire/Vari/chapitres-pdf/chapitre-3.pdf>. Acesso: 28 ago 2022.

pt:

- Estruturas de Controle estabelecem a ordem na qual as instruções do programa são executadas. Estas estruturas usam condições que são testadas a fim de determinar o fluxo de execução das instruções.

fonte:

- <https://sites.google.com/site/unipliprogramacaoestruturada/aulas/aula-3---estruturas-de-controle>. Acesso: 28 ago 2022.

contexto:**fr:**

- "Un langage relationnel n'est pas un langage de programmation : il ne possède pas les **structures de contrôle** de base d'un langage de programmation (condition, itération, ...)."

pt:

- "A combinação de espaçamento vertical antes e após as **estruturas de controle** e dos recuos do corpo dessas estruturas dá aos programas um aspecto bidimensional que aumenta muito a sua legibilidade."

structure de données//estrutura de dados**UTEs relacionadas:****fr:**

- structure de sélection
- structure de contrôle

pt:

- estrutura de seleção
- estrutura de controle

def:**fr:**

- Une structure de données est un format spécialisé pour organiser, traiter, récupérer et stocker des données. Bien qu'il existe plusieurs types de structures de base de donnée, toute structure de données est conçue pour organiser les données en fonction d'un objectif spécifique afin qu'elles puissent être accessibles et utilisées de manière appropriée.

source:

- <https://actualiteinformatique.fr/data/definition-structure-des-donnees-data-structure>. Acesso: 28 ago 2022.

pt:

- As estruturas de dados definem a organização, métodos de acesso e opções de processamento para a informação manipulada pelo programa.

fonte:

- <https://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node10.html>. Acesso: 28 ago 2022.

contexto:

fr:

- " Par contre, certains langages plus modernes comme l'Algol ou le Pascal et le C permettent de concevoir et de manipuler facilement des **structures de données** très complexes."

pt:

- "Analogamente, as variáveis compostas homogêneas são **estruturas de dados** que se caracterizam por um conjunto de variáveis do mesmo tipo."

système d'exploitation//sistema operacional

def:

fr:

- Le système d'exploitation (SE, en anglais Operating System ou OS) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications informatiques de l'utilisateur (traitement de texte, jeu vidéo...).

source:

- <https://www.techno-science.net/definition/7638.html>. Acesso: 26 ago 2022.

pt:

- Sistemas operacionais (SO) são agrupamentos de programas que gerenciam recursos, processadores, dispositivos e dados da máquina.

fonte:

- <https://www.inf.ufsc.br/~j.barreto/cca/sisop/sisoperac.html>. Acesso: 26 ago 2022.

contexto:

fr:

- "Le **système d'exploitation** permet de s'affranchir en apparence de cette limite et d'avoir ainsi plusieurs programmes qui s'exécutent en même temps."

pt:

- "O comando gettime retorna os valores da hora, minuto, segundo e centésimo de segundo da hora do **sistema operacional** e as variáveis que receberão esses valores devem ser do tipo word."

tableau à deux dimensions//matriz bidimensional

def:

fr:

- Un tableau à deux dimensions est un tableau multidimensionnel qui essaie de représenter des éléments sous forme de une table ou matrices. Il dispose de deux indices; une pour la ligne et l'autre pour la colonne.

source:

- HOUNGUE, P. Principes de Programmation. Université Virtuelle Africaine. Bénin. 2018. p.77.

pt:

- Compiladores são tradutores que convertem programas de linguagem de alto nível em linguagem de máquina.

fonte:

- <https://www.inf.pucrs.br/~gustavo/disciplinas/pli/material/paradigmas-aula08.pdf>

contexto:

fr:

- "Un tableau à n dimensions (un vecteur est représenté par un tableau à une dimension, une matrice par un **tableau à deux dimensions...**) est défini par n types d'indices séparés par des virgules."

pt:

- "A variável mat representa uma **matriz bidimensional** com 3 linhas e 5 colunas para armazenar números reais."

tableau//matriz

UTEs relacionadas:

fr:

- tableau à deux dimensions

pt:

- matriz bidimensional

def:

fr:

- Panneau sur lequel sont fixés des commutateurs, voyants, appareils de mesures, etc., servant à un opérateur pour diriger les opérations et les fonctions d'une machine.

source:

- <https://www.cnrtl.fr/definition/tableau>. Acesso: 26 ago 2022.

pt:

- Uma matriz pode ser definida como um conjunto de variáveis de mesmo tipo e identificadas pelo mesmo nome, diferenciadas por meio da especificação de suas posições dentro dessa estrutura.

fonte:

- AASCENCIO, A. F. G. ; CAMPOS, E. A. V. . Fundamentos da Programação de Computadores: Algoritmos, Pascal, C/C++ e Java. 3. ed. São Paulo: Pearson Education, 2012.

contexto:

fr:

- "Un élément du **tableau** est accédé en utilisant la variable du tableau suivie du numéro de l'élément appelé indice."

pt:

- "Os índices utilizados na linguagem JAVA para identificar as posições de uma **matriz** começam sempre em 0 (zero) e vão até o tamanho da dimensão menos uma unidade."

valeur d'une variable//valor de uma variável

def:

fr:

- Une fois la variable définie, une valeur lui est assignée, ou affectée.

source:

- http://hebergement.u-psud.fr/iut-orsay/Pedagogie/MPHY/Algo-C/Fiches_Cours.pdf.
Acesso: 29 ago 2022.

pt:

- As variáveis são inicializadas quanto ao tipo e quanto ao valor, isso significa que terá de ser definida qual o uso que iremos atribuir a ela.

fonte:

- <https://sites.usp.br/ltsi/variaveis/>. Acesso: 29 ago 2022.

contexto:

fr:

- "L'expression qui définit la **valeur d'une variable** peut faire référence à d'autres variables à condition que celles-ci aient été préalablement déclarées."

pt:

- "A diferença entre constantes e variáveis está no fato de que o valor de uma constante nunca se altera enquanto o **valor de uma variável** pode mudar"

variable globale//variável global

UTES relacionadas:

fr:

- variable locale

pt:

- variável local

def:

fr:

- Une variable devient globale dès qu'elle est définie en dehors de toute fonction, en général au début du fichier. Les variables globales sont utilisables par toutes les fonctions du fichier.

source:

- https://public.iutenligne.net/informatique/algorithmes-et-programmation/priou/LangageC/962_les_variables_globales.html. Acesso: 29 ago 2022.

pt:

- Uma variável é global se for definida fora de todas as funções do programa. As variáveis globais são acessíveis em qualquer ponto do programa (mesmo em outros módulos do programa).

fonte:

- <https://www.ime.usp.br/~pf/algoritmos/aulas/footnotes/global-variable.html>.
Acesso: 29 ago 2022.

contexto:**fr:**

- " Pour pouvoir affecter à une **variable globale** dans une fonction, cette variable doit faire l'objet d'une déclaration explicite comme variable globale de la forme `global variable_globale`."

pt:

- "Em geral, o uso de **variáveis globais** deve ser evitado, exceto em determinadas situações com exigências especiais de desempenho."

variable locale//variável local**UTEs relacionadas:****fr:**

- variable globale

pt:

- variável global

def:**fr:**

- Les variables locales sont les variables définies à l'intérieur du corps d'une fonction est dite locale à cette fonction, c'est-à-dire qu'elle n'est connue que de cette fonction.

fonte:

https://public.iutenligne.net/informatique/algorithmes-et-programmation/priou/LangageC/961_les_variables_locales.html. Acesso: 29 ago 2022.

pt:

- As variáveis locais são aquelas criadas dentro de uma função e só podem ser usadas dentro da função onde foram criadas, ou onde receberam o primeiro valor. Fora desta função a variável é desconhecida.

fonte:

- <https://www.ime.usp.br/~mms/mac1101s2019/aula12%20-%20Python%20-%20Funcoes,%20variaveis%20locais,%20globais%20e%20parametros.pdf>. Acesso: 29 ago 2022.

contexto:**fr:**

- "Un objet de champ persistant est en première approximation un moyen souple de stocker des informations de données, il est équivalent à une sorte de **variable locale** pour les données de la BD."

pt:

- "As **variáveis locais** declaradas no início de uma função possuem escopo de bloco, assim como os parâmetros da função, que são considerados variáveis locais por ida."

variable//variável

UTEs relacionadas:

fr:

- variable globale
- variable locale

pt:

- variável global
- variável local

def:

fr:

- La variable est utiliser pour stocker une information au cours d'un programme.

source:

- <http://cours.pise.info/algo/variables.html>. Acesso: 31 ago 2022.

pt:

- Variáveis são locais onde armazenamos valores na memória. Toda variável é caracterizada por um nome, que a identifica em um programa, e por um tipo, que determina o que pode ser armazenado naquela variável.

fonte:

- <https://www.ic.unicamp.br/~ffaria/lp1s2016/class02/lp-aula02.pdf>. Acesso: 31 ago 2022.

contexto:

fr:

- "Lors de l'appel du sous-programme, les contenus des **variables** paramètres sont transférés dans des variables locales, de mêmes types, et, lorsque l'on quitte le sous-programme, les contenus de ces variables sont automatiquement recopiés dans les variables originales."

pt:

- "Quando o valor da **variável** de controle indicar que o número correto de repetições foi realizado, o loop é encerrado e o computador continua a execução do programa a partir da instrução imediatamente após o loop."

Apêndice II – Lista de UTEs e palavras-chave.

| FR | PT-BR |
|-------------------|-------------------|
| affectation | atribuição |
| appel de fonction | chamada de função |
| arbre binaire | árvore binária |
| assembleur | compilador |
| base de données | banco de dados |

| | |
|--------------------------------|-----------------------------------|
| booléenne | booleano |
| boucle | laço |
| chaîne de caractère | cadeia de caractere |
| code machine | código de máquina |
| code source | código fonte |
| compilateur | compilador |
| débogueur | depurador |
| déclaration | declaração |
| déclaration de variable | declaração de variável |
| erreur de compilation | erro de compilação |
| exécuter des tâches | executar tarefas |
| exécuter un programme | executar um programa |
| fichier binaire | arquivo binário |
| fichier d'en-tête | arquivo de cabeçalho |
| fonction | função |
| fonction récursive | função recursiva |
| informatique | computação |
| informatique | informática |
| instruction | instrução |
| instruction d'affectation | instrução de atribuição |
| instruction en langage machine | instrução em linguagem de máquina |
| langage de bas niveau | linguagem de baixo nível |
| langage de haut niveau | linguagem de alto nível |
| mnémonique | mnemônico |
| opérateur | operador |
| opérateur d'affectation | operador de atribuição |
| opérateur d'incrémentation | operador de incremento |
| programmation orientée objet | programação orientada à objetos |
| programmation structurée | programação estruturada |
| récursive | recursivo |

| | |
|---------------------------|-----------------------|
| séquence d'échappement | sequência de escape |
| structure de contrôle | estrutura de controle |
| structure de données | estrutura de dados |
| système d'exploitation | sistema operacional |
| tableau | matriz |
| tableau à deux dimensions | matriz bidimensional |
| variable | variável |
| variables globales | variáveis globais |
| variables locales | variáveis locais |
| valeur d'une variable | valor de uma variável |

Apêndice III – Lista de UTEs extraordinárias extraídas do corpus em francês.

| |
|------------------------------------|
| FR |
| exécuter par une machine |
| exécuter un processeur |
| exécution séquentielle |
| expression booléenne |
| instruction de saut |
| instruction de saut conditionnel |
| instruction de saut inconditionnel |
| instruction de sélection |
| instruction* d'impression |
| opérateur de comparaison |
| opérateur de transtypage |
| point d'accès |
| programmation objet-orientée |
| programmation structurée |
| programmes utilitaires |

| |
|---|
| répartition |
| suite d'instructions |
| traducteurs de langage de programmation |
| traiter les données |
| transtypage |
| valeur booléenne |
| variable de type |
| variable pointant |
| fichier exécutable |
| séquence d'instruction |

Apêndice IV – Apêndice IV - Tradução espelhada de “Principes de Programmation”.

| Francês | Português |
|---|---|
| Université Virtuelle Africaine | Université Virtuelle Africaine |
| INFORMATIQUE APPLIQUÉE : | COMPUTAÇÃO APLICADA: |
| CSI 1204 | CSI 1204 |
| PRINCIPES DE | PRINCÍPIOS DE |
| PROGRAMMATION | PROGRAMAÇÃO |
| Pélagie Houngue | Pélagie Houngue |
| Unité 1. | Unidade 1. |
| Introduction à la programmation | Introdução à programação |
| Introduction à l'unité | Introdução à unidade |
| Les ordinateurs sont partout. | Os computadores estão em todos os lugares. |
| Que ce soit les grands magasins, les hôtels, les centrales atomiques ou à la défense. | Quer sejam lojas de departamento, hotéis, usinas nucleares ou na área de segurança. |
| Les ordinateurs aident les humains dans l'administration efficace d'une tâche. | Os computadores ajudam os humanos na administração eficiente de uma tarefa. |
| La croissance des ordinateurs est exponentielle et on n'a besoin de bons programmeurs. | O avanço dos computadores é exponencial e não são necessários bons programadores. |
| Par “bons programmeurs” nous n'entendons pas les programmeurs qui peuvent donner des solutions, mais les programmeurs qui peuvent donner des solutions efficaces. | Por “bons programadores”, não se quer dizer programadores que podem dar soluções, mas programadores que podem fornecer soluções eficazes. |
| Les bons programmeurs ne sont pas nés bons, au contraire, ils ont été formés. | Bons programadores não nascem bons, pelo contrário, eles foram treinados. |
| “Les compétences d'un programmeur se révèlent bien plus avec l'expérience et la pratique”, Harsha Priya, R. Ranjeet. | “As habilidades de um programador são reveladas muito mais com experiência e prática”, Harsha Priya, R. Ranjeet. |
| Dans cette unité, vous allez apprendre les concepts de base de la programmation et par la même | Nesta unidade, você aprenderá os conceitos básicos de programação e, ao mesmo tempo, |

| | |
|---|---|
| occasion étudierez l'histoire des langages de programmation. | estudará a história das linguagens de programação. |
| Objectifs de l'unité | Objetivos da unidade |
| À la fin de cette unité, vous devriez être capable de : | Ao final desta unidade, você deverá ser capaz de: |
| Expliquez ce qu'est la programmation et de définir les concepts de base de la programmation | Explicar o que é programação e definir os conceitos básicos de programação |
| Décrire l'évolution des langages de programmation et leurs influences sur l'évolution de la conception des langages | Descrever a evolução das linguagens de programação e suas influências na evolução do design da linguagem |
| Décrire les niveaux et les générations de langages de programmation | Descrever os níveis e gerações de linguagens de programação |
| Décrire les caractéristiques d'un bon langage | Descrever as características de uma boa linguagem de programação |
| Décrire et expliquer les caractéristiques des paradigmes des langages de programmation | Descrever e explicar as características dos paradigmas da linguagem de programação |
| Principes de Programmation | |
| Unité 1. | Unidade 1. |
| Introduction à la programmation | Introdução à programação |
| Termes clés | Termos-chave |
| La programmation est l'activité d'écriture d'instructions devant être exécutée par l'ordinateur. | Programação é a atividade de escrever instruções a serem executadas pelo computador. |
| Ces instructions indiquent à l'ordinateur de faire quelque chose. | Essas instruções dizem ao computador para executar uma ação. |
| Par exemple, résoudre un problème particulier, lire les notes des étudiants et calculer le rang par étudiant et de fournir une liste d'étudiants qui passent en classe supérieure et une autre liste des étudiants qui recommenceront le cours. | Por exemplo, resolva um problema específico, leia as notas dos alunos, calcule a classificação por aluno e forneça uma lista de alunos que passarão para a próxima série e outra lista de alunos que repetirão o curso. |
| Tout cela, ce sont des instructions (logiques). | Todas essas são instruções (lógicas). |
| Un ordinateur n'est pas en mesure d'agir sans instructions. | Um computador não é capaz de agir sem instruções. |
| Aussi, la programmation est l'art d'exécuter des tâches de calcul et de convertir ces tâches sous une forme lisible par la machine (Harsha Priya, R. Ranjeet) | Do mesmo modo, a programação é a arte de realizar tarefas computacionais e converter tais tarefas numa forma legível por máquina (Harsha Priya, R. Ranjeet). |
| Le langage de programmation est une série d'instructions pour écrire des programmes. | A linguagem de programação é uma série de instruções para escrever programas. |
| Les paradigmes des langages de programmation définissent la manière dont les programmes sont structurés ou les styles de programmation. | Os paradigmas da linguagem de programação definem como os programas ou os estilos de programação são estruturados |
| Un programmeur est la personne qui écrit des instructions ou développe un programme en utilisant la programmation. | Um programador é a pessoa que escreve instruções ou desenvolve um programa usando a programação. |
| Un programme est une série d'instructions qui demande à l'ordinateur de faire quelque chose. | Um programa é uma série de instruções que dizem ao computador para realizar alguma ação. |
| C'est le produit final de l'activité de programmation, aussi connu sous le nom de logiciel. | É o produto da atividade de programação, também conhecida como software. |

| | |
|--|--|
| La syntaxe d'un langage informatique est l'ensemble de règles qui définit les combinaisons de symboles qui sont considérés comme un document ou un fragment correctement structuré dans cette langue (http:// en.wikipedia.org). | A sintaxe de uma linguagem de computador é o conjunto de regras que definem as combinações de símbolos que são considerados um documento ou fragmento adequadamente estruturado nessa linguagem (http://pt.wikipedia.org). |
| La sémantique définit le sens syntaxique légal des chaînes de caractères définis par un langage de programmation spécifique, montrant le calcul implicé (http://en.wikipedia.org). | A semântica define o significado sintático legal das cadeias de caracteres definidas por uma linguagem de programação específica, mostrando o cálculo envolvido (http://pt.wikipedia.org). |
| Activités d'apprentissage | Atividades de aprendizagem |
| Activité 1.1 - Histoire de la programmation | Atividade 1.1 - História da Programação |
| Introduction | Introdução |
| La programmation peut être définie comme l'action d'utiliser un langage de programmation pour écrire des instructions qui peuvent être exécutées par l'ordinateur dans le but de résoudre des problèmes en appliquant correctement la syntaxe et la sémantique du langage. | A programação pode ser definida como a ação de usar uma linguagem de programação para escrever instruções que podem ser executadas pelo computador com o objetivo de resolver problemas aplicando corretamente a sintaxe e a semântica da linguagem. |
| Des centaines de langage de programmation ont été développés et certains existent encore tandis que d'autres n'existent plus. | Centenas de linguagens de programação foram desenvolvidas e algumas ainda existem, enquanto outras não existem mais. |
| Cela signifie que le développement des langages de programmation a parcouru un long chemin et a ainsi une longue histoire ; de Plankalkül, Prolog, FORTRAN, C, Visual Basic.Net, à Rust et Swift. | Isso significa que o desenvolvimento de linguagens de programação percorreu um longo caminho e, portanto, tem uma longa história; de Plankalkül, Prolog, FORTRAN, C, Visual Basic.Net, a Rust e Swift. |
| Certains de ces langages ont été améliorés et ont gagné plus de popularité que d'autres. | Algumas dessas linguagens foram aprimoradas e ganharam mais popularidade do que outras. |
| Ci-dessous, nous avons un résumé de l'histoire des langages de programmation tel que présentée par Seema Kedar. | Abaixo, temos um resumo da história das linguagens de programação, como apresentado por Seema Kedar. |
| 1951-1955 : | 1951-1955 : |
| Utilisation expérimentale des compilateurs d'expression. | Uso experimental de compiladores de expressão. |
| 1956-1960 : | 1956-1960 : |
| FORTRAN, COBOL, LISP, Algol 60. | FORTRAN, COBOL, LISP, Algol 60. |
| 1961-1965 : la notation APL, Algol 60 (révisée), SNOBOL, CPL. | 1961-1965: a notação APL, Algol 60 (revisado), SNOBOL, CPL. |
| 1966-1970 : | 1966-1970 : |
| APL, SNOBOL 4, FORTRAN 66, BASIC, SIMULA, Algol 68, Algol-W, BCPL. | APL, SNOBOL 4, FORTRAN 66, BÁSICO, SIMULA, Algol 68, Algol-W, BCPL. |
| 1971-1975 : | 1971-1975 : |
| Pascal, PL/1 (Standard), C, Scheme, Prolog. | Pascal, PL/1 (Padrão), C, Esquema, Prólogo. |
| 1976-1980 : | 1976-1980 : |
| Smalltalk, Ada, Fortran 77, ML. | Smalltalk, Ada, Fortran 77, ML. |
| Détails de l'activité | Detalhes da atividade |
| Dans cette activité, vous devez lire l'histoire de la programmation, c'est-à-dire comment la programmation a commencé et l'application au programme, leur descente vers les langages | Nesta atividade, você deve ler a história da programação, ou seja, como a programação começou e a aplicação ao programa, sua descida para as linguagens atuais, as influências na |

| | |
|---|--|
| actuels, les influences sur l'évolution de la conception et le développement des langages. | evolução do design e desenvolvimento de linguagens. |
| Les informations peuvent être obtenues à partir de "Programming Paradigms and Methodology" de Seema Kedar et "Programming and PROB using C" de ISRD et d'autres références énumérées dans la section référence de cette unité et des informations connexes et pertinentes disponibles sur Internet. | As informações podem ser obtidas em "Paradigmas e Metodologia de Programação" de Seema Kedar e "Programação e PROB usando C" do ISRD e outras referências listadas na seção de referência desta unidade e informações relacionadas e relevantes disponíveis na Internet. |
| Les apprenants devraient également compléter les lectures en faisant des recherches individuelles afin d'améliorer leur compréhension. | Os alunos também devem concluir as leituras fazendo pesquisas individuais para melhorar sua compreensão. |
| Dans cette activité, vous devez: | Nesta atividade, você precisa: |
| Lister au moins cinq problèmes qui peuvent être résolus à l'aide d'un ordinateur (juste pour vérifier la compréhension de ce qu'est la programmation) | Liste pelo menos cinco problemas que podem ser resolvidos usando um computador (a fim de verificar o entendimento sobre o que é programação) |
| Décrire comment la programmation a commencé et les situations qui ont conduit au développement des programmes | Descrever como a programação começou e as situações que levaram ao desenvolvimento do programa |
| En utilisant un schéma, décrire la prolifération des langages de programmation; Nommer et décrire les utilisations et la signification des langages de programmation. | Usando um diagrama, descreva a proliferação de linguagens de programação; Nomeie e descreva os usos e significados das linguagens de programação. |
| Montrer et décrire tous les descendants d'un langage particulier. | Mostre e descreva todos os descendentes de uma linguagem específica. |
| Conclusion | Conclusão |
| Cette activité aidera les apprenants à apprécier l'histoire de la programmation et peut également stimuler les apprenants à comprendre comment tirer parti des situations et des environnements tels qu'ils se présentent pour proposer et concevoir de nouveaux ou d'améliorer les langages existants. | Esta atividade ajudará os alunos a valorizar a história da programação como também pode estimulá-los a entender como tirar proveito de situações e ambientes à medida que se apresentam para propor e projetar novas linguagens ou aprimorar as existentes. |
| Évaluation | Avaliação |
| Vous pouvez faire cette évaluation individuellement ou en groupe avant d'essayer de faire correctement les évaluations sommatives. | Você pode fazer esta avaliação individualmente ou em grupo, antes de tentar fazer as avaliações cumulativas corretas. |
| Cette évaluation porte sur l'activité qui vient de s'achever. | Essa avaliação se concentra na atividade que acaba de ser concluída. |
| Selon ce que vous savez sur l'histoire de l'informatique, quand a été conçu le premier langage de programmation et quelle était son utilisation? | Com base no que você sabe sobre a história da ciência da computação, quando foi projetada a primeira linguagem de programação e para que foi sua utilidade? |
| Quelle est l'importance d'étudier l'histoire de la programmation ? | Qual é a importância de estudar a história da programação? |
| Qu'est ce qui a influencé le développement des langages de programmation C et Fortran ? | O que influenciou o desenvolvimento das linguagens de programação C e Fortran? |
| Maintenant, vous êtes au courant que des centaines de langages de programmation existent. | Agora você está ciente de que existem centenas de linguagens de programação. |

| | |
|---|---|
| Certains sont largement utilisés dans l'industrie aujourd'hui et certains ne sont jamais du tout mentionnés ou sont utilisés par un petit groupe de programmeurs. | Alguns são amplamente usados na indústria hoje e alguns nunca são mencionados ou são usados por um pequeno grupo de programadores. |
| Pourquoi pensez-vous que la plupart de ces langages ont gagné plus de popularité que d'autres? | Por que você acha que a maioria dessas linguagens ganhou mais popularidade do que outros? |
| Activité 1.2 - Les niveaux, les générations et les paradigmes des langages de programmation | Atividade 1.2 - Níveis, Gerações e Paradigmas das Linguagens de Programação |
| Présentation | Apresentação |
| Les programmeurs utilisent le langage de programmation pour écrire des programmes et les langages de programmation ont évolué au fil du temps avec les premiers langages connus également comme langages de première génération utilisant le code machine pour programmer l'ordinateur datant de 1940 à 1950. | Os programadores usam a linguagem de programação para escrever programas e as linguagens de programação evoluíram ao longo do tempo, com as primeiras linguagens também conhecidas como linguagens de primeira geração usando código de máquina para programar o computador datando de 1940 a 1950. |
| Ce langage est qualifié de langage de bas niveau. | Essa linguagem é conhecida como linguagem de baixo nível. |
| Le développement suivant a été les langages de deuxième génération datant de la période de 1950 à 1958 qui utilisent le langage d'assemblage pour représenter des instructions en langage machine. | O próximo desenvolvimento foram as linguagens de segunda geração que datam do período de 1950 a 1958 que usam a linguagem Assembly para representar instruções em linguagem de máquina. |
| Elles sont ensuite traduites en code machine par un assembleur. | Elas são então traduzidas em código de máquina por um montador. |
| Ce langage était aussi un langage de bas niveau. | Essa linguagem também era de baixo nível. |
| Ensuite, les langages de troisième génération qui inclut le développement des langages de programmation de haut niveau tels que C, Pascal, FORTRAN et COBOL, et datant de la période 1958 à 1985. | Em seguida, as linguagens de terceira geração que incluem o desenvolvimento de linguagens de programação de alto nível, como C, Pascal, FORTRAN e COBOL, e datam do período de 1958 a 1985. |
| Ils étaient plus faciles à utiliser que les langages d'assemblage et le code machine et ont contribué à améliorer la qualité et la productivité. | Eles eram mais fáceis de usar do que linguagens de montagem e código de máquina e ajudaram a melhorar a qualidade e a produtividade. |
| Les compilateurs et les interpréteurs sont utilisés pour traduire les instructions d'un langage de haut niveau vers le langage machine. | Compiladores e intérpretes são usados para traduzir instruções de uma linguagem de alto nível para a linguagem de máquina. |
| Unité 1. | |
| Introduction à la programmation | |
| Unité 1. | |
| Introduction à la programmation | |
| Principes de Programmation | |
| Principes de Programmation | |
| Il y a également une quatrième génération de langages à partir de 1985. | Há também uma quarta geração de linguagens a partir de 1985. |
| Ces langages incluent les générateurs de rapports, qui réduisent ainsi l'effort de programmation. | Essas linguagens incluem geradores de relatórios, que reduzem o esforço de programação. |
| Les langages de programmation de cinquième génération datent à partir de 1990 et sont | As linguagens de programação de quinta geração datam de 1990 e são usadas principalmente no campo da inteligência artificial. |

| | |
|--|---|
| principalement utilisés dans le domaine de l'intelligence artificielle. | |
| Chaque génération a ses propres caractéristiques et une nouvelle génération de langage est plus améliorée qu'une ancienne. | Cada geração tem suas próprias características e uma nova geração de linguagem é mais aprimorada do que uma antiga. |
| Un langage de programmation s'inscrit dans un paradigme particulier ou supporte plus d'un paradigme pour concevoir et mettre en œuvre un programme. | Uma linguagem de programação se encaixa em um paradigma específico ou suporta mais de um paradigma para projetar e implementar um programa. |
| Le paradigme définit la façon dont le programme est structuré. | O paradigma define como o programa é estruturado. |
| La figure ci-dessous illustre les niveaux de langages de programmation. | A figura abaixo ilustra os níveis da linguagem de programação. |
| Langages de Programmation | Linguagens de programação |
| Langages haut niveau | Linguagens de alto nível |
| Langage machine | Linguagem de máquina |
| Langage | Linguagem |
| d'assemblage | Assembly |
| Figure 1.1 : | Figura 1.1: |
| Les niveaux de langages de programmation | Níveis de linguagens de programação |
| (Source : | (Fonte: |
| ISRD) | ISRD) |
| Le schéma montre qu'il y a deux grands niveaux de langages de programmation : les langages de programmation de bas niveau et de haut niveau. | O diagrama mostra que existem dois níveis principais de linguagens de programação: linguagens de programação de baixo e alto nível. |
| Langage machine : | Linguagem da máquina: |
| En utilisant ce langage, les programmes sont écrits en utilisant des 0 et des 1. | Usando essa linguagem, os programas são escritos usando 0s e 1s. |
| Il a été utilisé au départ pour programmer un ordinateur. | Foi originalmente usado para programar um computador. |
| Il est plus rapide parce que les instructions sont directement exécutables, et il fait un usage plus efficace des ressources comme les registres et les unités de stockage. | É mais rápido porque as instruções são executáveis diretamente e fazem uso mais eficiente de recursos, como registros e unidades de armazenamento. |
| Actuellement, ce langage n'est pas très apprécié parce que les programmes ne sont pas portables, c'est-à-dire ils sont dépendants de la machine, il est plus sujet à des erreurs et difficile à déboguer, il nécessite des compétences de programmation de haut niveau et ainsi un coût élevé pour la formation. | Atualmente, essa linguagem não é muito popular porque os programas não são portáteis, ou seja, são dependentes da máquina, são mais propensos a erros e difíceis de depurar, requer habilidades de programação de alto nível e, portanto, um alto custo para treinamento. |
| Langage d'assemblage : | Linguagem Assembly: |
| Ce langage utilise des mnémoniques ou de courtes abréviations représentant une instruction. | Essa linguagem usa mnemônicos ou abreviações curtas que representam uma instrução. |
| Il est plus facile à utiliser que le langage machine simplement parce les mnémoniques sont plus proches du programmeur que de l'ordinateur (machine). | É mais fácil de usar do que a linguagem de máquina simplesmente porque os mnemônicos estão mais próximos do programador do que do computador (máquina). |
| L'assembleur est utilisé pour convertir les instructions en langage machine. | O Assembler é usado para converter instruções em linguagem de máquina. |

| | |
|--|---|
| Langages de haut niveau : | Linguagens de alto nível: |
| Les programmes sont écrits en anglais comme des déclarations. | Os programas são escritos em inglês como declarações. |
| Ils ne sont pas directement exécutables donc les traducteurs qui comprennent les compilateurs et les interpréteurs sont utilisés pour convertir les instructions en langage machine. | Eles não são executáveis diretamente, então tradutores que entendem compiladores e intérpretes são usados para converter instruções em linguagem de máquina. |
| Les langages qui appartiennent à cette catégorie sont très appréciés car ils sont portables, faciles à apprendre et pour écrire un programme, la disponibilité des bibliothèques, la facilité de maintenance et de documentation. | As linguagens incluídas nesta categoria são altamente valorizadas por sua portabilidade, facilidade de aprendizagem e escrita de um programa, disponibilidade de bibliotecas, facilidade de manutenção e documentação. |
| Les langages de haut niveau ont différents attributs ou caractéristiques qui comprennent : la portabilité, la lisibilité, le support de la concurrence, le support du mixage de langage, la fiabilité, la modularité, le support du temps réel et l'orthogonalité. | Linguagens de alto nível têm diferentes atributos ou características que incluem: portabilidade, legibilidade, suporte à competição, suporte à mistura de linguagens, confiabilidade, modularidade, suporte em tempo real e ortogonalidade. |
| Langages | Linguagens |
| procéduraux | procedimentais |
| Langages de 4ème | Linguagens de 4ª |
| génération | geração |
| Langages de | Linguagens de |
| programmation | programação |
| visuelle | visual |
| Langages bas niveau | Linguagens de baixo nível |
| Langages de quatrième génération : | Linguagens de quarta geração: |
| Ces langages sont aussi connus comme 4GL et soulignent sur ce qui doit être accompli plutôt que la façon de l'accomplir. | Essas linguagens também são conhecidas como 4GL e enfatizam o que precisa ser realizado e não como realizá-lo. |
| Comme exemple nous avons Oracle, SQL, etc. Ils sont principalement utilisés pour accéder aux bases de données. | Como exemplo, temos Oracle, SQL, etc. Eles são usados principalmente para acessar bancos de dados. |
| Les 4GL augmentent la productivité. | O 4GL aumenta a produtividade. |
| Ils ont diverses caractéristiques qui les rendent attrayants. | Eles têm várias características que os tornam atraentes. |
| Il s'agit notamment de : | Isso inclui: |
| la facilité d'utilisation | facilidade de uso |
| le nombre restreint de fonctions | o número limitado de funções |
| la disponibilité d'options | a disponibilidade de opções |
| les options par défaut. | as opções padrão. |
| Les compilateurs | Compiladores |
| Un compilateur traduit un programme de haut niveau en langage machine. | Um compilador traduz um programa de alto nível em linguagem de máquina. |
| Le programme de haut niveau est appelé code source, qui se traduit pour produire un code lisible par la machine connu sous le nom de code objet. | O programa de alto nível é chamado de código-fonte, que se traduz em produzir código legível por máquina conhecido como código objeto. |
| La plupart ou tous les langages de haut niveau utilisent un compilateur. | A maioria ou todas as linguagens de alto nível usam um compilador. |
| Par exemple, le compilateur C, C++, etc. | Por exemplo, o compilador C, C++, etc. |

| Les Interpréteurs | Os intérpretes |
|---|--|
| Tout comme les compilateurs, les interpréteurs traduisent le programme écrit dans un langage de haut niveau en un format lisible par la machine. | Assim como os compiladores, os intérpretes traduzem o programa escrito em uma linguagem de alto nível em um formato legível por máquina. |
| La différence avec les compilateurs, c'est que, les interpréteurs traduisent le code déclaration par déclaration et si une erreur est rencontrée, il s'arrête et poursuit après que l'erreur ait été corrigée. | A diferença com os compiladores é que os intérpretes traduzem declaração de código por declaração e, se um erro for encontrado, ele pára e continua após o erro ter sido corrigido. |
| Alors que les compilateurs traduisent l'ensemble du programme, et ensuite listent les erreurs, le cas échéant. | Enquanto os compiladores traduzem o programa inteiro e, em seguida, listam os erros, se houver. |
| Cela signifie que les compilateurs sont plus rapides que les interpréteurs. | Isso significa que os compiladores são mais rápidos que os intérpretes. |
| Influences sur l'évolution de la conception des langages | Influências na evolução do design de linguagem |
| Seema Kedar a présenté six influences : | Seema Kedar apresentou seis influências: |
| Capacités des ordinateurs : | Capacidades do computador: |
| Les ordinateurs ont grandement amélioré la façon dont ils traitent l'information. | Os computadores melhoraram muito a maneira como processam informações. |
| Il y a eu des améliorations majeures sur le matériel et dans le même temps au niveau logiciel (systèmes d'exploitation). | Houve grandes melhorias no hardware e, ao mesmo tempo, no nível do software (sistemas operacionais). |
| Ces améliorations participent à la conception de meilleurs langages. | Essas melhorias contribuem para o design de linguagens melhores. |
| Applications : | Aplicações: |
| Application signifie l'utilisation des ordinateurs. | Aplicação significa o uso de computadores. |
| Ils ont commencé par faire de petits calculs, en manipulant des activités militaires, des activités scientifiques, comme la recherche, la médecine, des affaires aux jeux et l'Internet. | Eles começaram fazendo pequenos cálculos, manipulando atividades militares, atividades científicas, como pesquisa, medicina, negócios para jogos e a Internet. |
| Les exigences des nouveaux domaines d'application conduisent à l'amélioration des langages existants ou à la conception de nouveaux langages | Os requisitos dos novos campos de aplicação levam à melhoria dos linguagens existentes ou ao design de novos linguagens. |
| Méthodes de programmation : | Métodos de programação: |
| Comme les problèmes deviennent plus complexes, de nouvelles méthodes sont inventées pour aider à réduire la complexité et améliorer la productivité. | À medida que os problemas se tornam mais complexos, novos métodos são inventados para ajudar a reduzir a complexidade e melhorar a produtividade. |
| Méthodes d'implémentation : | Métodos de implementação: |
| Le développement de meilleures méthodes de mise en œuvre a affecté le choix des caractéristiques à inclure dans les nouveaux modèles de langage | O desenvolvimento de melhores métodos de implementação afetou a escolha de recursos a serem incluídos em novos modelos de linguagem |
| Études théoriques : | Estudos teóricos: |
| La recherche dans la base conceptuelle pour la conception et la mise en œuvre des langages en utilisant des méthodes mathématiques formelles a augmenté notre compréhension des forces et faiblesses des fonctionnalités des langages qui ont | A pesquisa sobre a base conceitual para o design e implementação da linguagem usando métodos matemáticos formais aumentou nossa compreensão dos pontos fortes e fracos dos recursos da linguagem que influenciaram a |

| | |
|--|--|
| influencé l'inclusion de ces fonctionnalités dans la conception de nouveau langage | inclusão desses recursos no novo design de linguagem. |
| Normalisation : | Padronização |
| La nécessité d'un langage standard qui peut être facilement mis en œuvre en utilisant un système d'ordinateurs permettant aux programmes d'être traduits d'un ordinateur à un autre, a exercé une forte influence sur l'évolution de la conception des langages. | A necessidade de uma linguagem padrão que possa ser facilmente implementada usando um sistema de computador que permita que os programas sejam traduzidos de um computador para outro, teve uma forte influência na evolução do design da linguagem. |
| Les langages ont différentes forces. | As linguagens têm pontos fortes diferentes. |
| Certains langages soutiennent des applications particulières que d'autres. | Algumas linguagens oferecem suporte a aplicativos específicos do que outros. |
| Qu'est-ce donc, définir un bon langage de programmation ? | Então, o que significa definir uma boa linguagem de programação? |
| Clarté, simplicité et unité | Clareza, simplicidade e união |
| Orthogonalité | Ortogonalidade |
| Naturalité des applications | Naturalidade das aplicações |
| Support pour l'abstraction | Suporte para abstração |
| Facilité de vérification du programme | Facilidade de verificar o programa |
| Environnement de programmation | Ambiente de programação |
| Portabilité des programmes | Portabilidade do programa |
| coût | custo |
| coût de la formation | custo do treinamento |
| coût de l'écriture du programme | custo de escrever o programa |
| coût de l'exécution du programme | custo de execução do programa |
| coût de la traduction | custo de tradução |
| coût de mise en œuvre | custo de implementação |
| coût de l'entretien | custo de manutenção |
| Les paradigmes des langages de programmation | Os paradigmas das linguagens de programação |
| Les langages de programmation sont structurés de différentes manières. | As linguagens de programação são estruturadas de maneiras diferentes. |
| Par conséquent les langages peuvent être regroupés en fonction de la façon dont ils sont structurés ou conçus, simplement connue comme paradigme. | Portanto, as linguagens podem ser agrupadas de acordo com a forma como são estruturadas ou projetadas, simplesmente conhecidas como paradigma. |
| Les paradigmes comprennent ; | Os paradigmas incluem; |
| Langages impératifs ou procéduraux : | Linguagens obrigatórias ou processuais: |
| Ces langages sont basés sur des commandes et chaque exécution d'une commande modifie les valeurs dans la mémoire. | Essas linguagens são baseadas em comando e cada execução de um comando altera os valores na memória. |
| L'exécution des commandes est séquentielle. | O atendimento do pedido é sequencial. |
| Langages fonctionnels : | Linguagens funcionais: |
| Ce paradigme est orienté fonction. | Esse paradigma é orientado para a função. |
| Il ne met pas l'accent sur la séquence des événements qui conduisent à des changements de valeurs de mémoire, mais plutôt s'intéresse à ce que la déclaration peut faire. | Ele não se concentra na sequência de eventos que levam a mudanças nos valores da memória, mas sim no que a declaração pode fazer. |
| Langages logiques ou basés sur les règles : | Linguagens lógicas ou baseadas em regras: |
| Ils supportent la prise de décision basée sur les conditions fournies. | Eles apoiam a tomada de decisões com base nas condições fornecidas. |

| | |
|--|--|
| Langages orientés objet : | Linguagens orientadas a objetos: |
| C'est le paradigme qui fournit des solutions qui représentent le scénario du monde réel. | É o paradigma que fornece soluções que representam o cenário do mundo real. |
| Il met l'accent davantage sur des données. | Ele coloca mais ênfase nos dados. |
| Il utilise le concept de classes où un objet est considéré en termes de données (ce qu'il peut traiter) et ce qu'il peut faire (méthodes). | Ele usa o conceito de classes onde um objeto é considerado em termos de dados (o que ele pode processar) e o que ele pode fazer (métodos). |
| Langages concurrents : | Linguagens concorrentes: |
| Ils appliquent le concept d'un processus. | Eles aplicam o conceito de um processo. |
| Un processus correspond à un calcul séquentiel, avec son propre thread de contrôle. | Um processo corresponde a um cálculo sequencial, com seu próprio segmento de controle. |
| Détails de l'activité | |
| Dans cette activité, vous devez lire sur les niveaux (d'autres auteurs appellent cela types de langages de programmation) et les générations de langages de programmation; constater les caractéristiques, les avantages et les inconvénients de chaque type/niveau, y compris les générations de langages de programmation. | Nesta atividade, você deve ler sobre níveis (outros autores chamam esse tipo de linguagens de programação) e gerações de linguagens de programação; veja as características, vantagens e desvantagens de cada tipo/nível, incluindo gerações de linguagens de programação. |
| Il faut lire sur les caractéristiques d'un bon langage de programmation. | Você precisa ler sobre as características de uma boa linguagem de programação. |
| Il faut lire également sur le paradigme des langages de programmation; les caractéristiques de chaque paradigme y compris les langages qui correspondent à chaque paradigme. | Também devemos ler sobre o paradigma das linguagens de programação; as características de cada paradigme, incluindo as linguagens que correspondem a cada paradigme. |
| Nous recommandons "Programming Paradigms and Methodology" écrit par Seema Kedar et "Programming and PROB using C" écrit par ISRD. | Recomendamos "Programming Paradigms and Methodology" escrito por Seema Kedar e "Programming and PROB using C" escrito por ISRD. |
| Après les avoir lu, écrire de brefs résumés. | Depois de lê-los, escreva breves resumos. |
| Décrire les niveaux et les générations de langages de programmation et citer les langages de programmation qui appartiennent à chaque niveau et génération. | Descreva os níveis e gerações de linguagens de programação e liste as linguagens de programação que pertencem a cada nível e geração. |
| Décrire les caractéristiques des langages de programmation qui appartiennent à un niveau et génération particulier. | Descreva as características das linguagens de programação que pertencem a um determinado nível e geração. |
| Décrire les caractéristiques d'un bon langage de programmation. | Descreva as características de uma boa linguagem de programação. |
| Comment un bon langage de programmation se définit? | Como uma boa linguagem de programação é definida? |
| Décrire les caractéristiques des langages de programmation qui sont regroupés au sein d'un paradigme de programmation particulier. | Descreva as características das linguagens de programação agrupadas dentro de um paradigma de programação específico. |
| Activité pratique | Atividade prática |
| Manuellement ou en utilisant un logiciel, élaborer un schéma qui regroupe tous les langages de programmation identifiés dans l'activité 1.1 selon le niveau et la génération de langages de programmation correspondant. | Manualmente ou usando software, desenvolva um diagrama que agrupe todas as linguagens de programação identificadas na Atividade 1.1 de acordo com o nível e a geração correspondente de linguagens de programação. |

| | |
|--|---|
| Manuellement ou en utilisant un logiciel, élaborer un schéma qui regroupe les langages de programmation identifiés dans l'activité 1.1 et selon la catégorie de paradigme de programmation à laquelle ils appartiennent. | Manualmente ou usando software, desenvolva um diagrama que agrupe as linguagens de programação identificadas na Atividade 1.1 e de acordo com a categoria de paradigma de programação à qual pertencem. |
| Conclusion | Conclusão |
| A ce stade du cours, les apprenants doivent avoir un bon niveau de connaissances sur les caractéristiques des niveaux, des générations et des paradigmes des langages de programmation. | Nesta fase do curso, os alunos devem ter um bom nível de conhecimento sobre as características dos níveis, gerações e paradigmas das linguagens de programação. |
| Ils devraient être en mesure de décrire les langages de programmation qui appartiennent à chaque niveau, génération et paradigme. | Eles devem ser capazes de descrever as linguagens de programação que pertencem a cada nível, geração e paradigma. |
| Résumé de l'unité | Resumo da unidade |
| Cette unité a permis d'apprendre ce qu'est la programmation, l'histoire de la programmation, les niveaux, les générations et les paradigmes des langages de programmation. | Esta unidade permitiu aprender o que é programação, a história da programação, os níveis, as gerações e os paradigmas das linguagens de programação. |
| Jusqu'ici, l'apprenant comprend les concepts de base de la programmation et est prêt à passer à l'unité suivante. | Até agora, o aluno entende os conceitos básicos de programação e está pronto para passar para a próxima unidade. |
| Évaluation | Avaliação |
| Vous pouvez faire cette évaluation individuellement ou en groupe avant de tenter de remplir correctement les évaluations formatives. | Você pode concluir essa avaliação individualmente ou em grupo antes de tentar concluir as avaliações formativas corretamente. |
| Cette évaluation est sur l'activité qui vient de s'achever. | Essa avaliação é baseada na atividade que acabou de ser concluída. |
| Le langage machine est connu comme un langage de bas niveau et même si l'assembleur utilise des mnémoniques, il est toujours considéré comme un langage de bas niveau. | A linguagem de máquina é conhecida como linguagem de baixo nível e, embora o montador use mnemônicos, ainda é considerada uma linguagem de baixo nível. |
| Pourquoi pensez-vous cela? | Por que você acha isso? |
| Les langages de programmation sont classés suivant des générations. | As linguagens de programação são classificadas por gerações. |
| Pourquoi? | Por quê? |
| Décrire au moins quatre caractéristiques d'un bon langage de programmation. | Descreva pelo menos quatro características de uma boa linguagem de programação. |
| Le Langage de haut niveau peut être traduit en langage machine en utilisant soit un compilateur ou un interpréteur. | A linguagem de alto nível pode ser traduzida para a linguagem de máquina usando um compilador ou um intérprete. |
| Quels sont les avantages et les inconvénients d'un compilateur et d'un interpréteur. | Quais são as vantagens e desvantagens de um compilador e um intérprete. |
| Lequel des deux traducteurs est plus efficace et pourquoi? | Qual dos dois tradutores é mais eficiente e por quê? |
| Décrire les cinq générations de langages de programmation. | Descreva as cinco gerações de linguagens de programação. |
| Qu'est ce qui a influencé le développement des paradigmes de programmation? | O que influenciou o desenvolvimento dos paradigmas de programação? |

| | |
|---|--|
| Que pensez-vous que sera l'avenir des langages de programmation. | O que você acha que será o futuro das linguagens de programação. |
| Pensez-vous que nous aurons plus de générations et de paradigmes de langages de programmation? | Você acha que teremos mais gerações e paradigmas de linguagens de programação? |
| Principes de Programmation | |
| Unité 1. | |
| Introduction à la programmation | |
| Activité 1.3 -Activité en équipe (Collaboration) | Atividade 1.3 - Atividade da equipe (colaboração) |
| Introduction | Introdução |
| Cette activité implique une activité de groupe où les apprenants seront amenés à comprendre la pratique des langages de programmation et des paradigmes. | Esta atividade envolve uma atividade em grupo em que os alunos serão levados a entender a prática de linguagens e paradigmas de programação. |
| Les apprenants sont tenus de faire une recherche à partir des développeurs individuels ou des entreprises sur les langages couramment utilisés et les paradigmes supportés par les langages. | Os alunos devem pesquisar com desenvolvedores ou empresas sobre linguagens comumente usadas e os paradigmas suportados pelas linguagens. |
| Détails de l'activité | Detalhes da atividade |
| En visitant les développeurs individuels et / ou des entreprises : | Ao visitar desenvolvedores e/ou empresas individuais: |
| Donner les langages de programmation les plus populaires dans l'industrie et les paradigmes supportés par les langages. | Forneça as linguagens de programação mais populares do setor e os paradigmas suportados pelas linguagens. |
| Donnez les raisons de leur popularité. | Dê razões para sua popularidade. |
| Dire comment les programmeurs sélectionnent le langage de programmation à utiliser dans la résolution d'un problème. | Diga como os programadores selecionam a linguagem de programação a ser usada na solução de um problema. |
| Conclusion | Conclusão |
| Le but de cette activité est d'élargir les capacités d'innovation des apprenants et à la fin l'apprenant devrait être en mesure d'apprécier les utilisations de divers langages et paradigmes et être capable de travailler en collaboration avec les autres. | O objetivo desta atividade é ampliar as capacidades inovadoras dos alunos e, no final, o aluno deve ser capaz de apreciar os usos de várias línguas e paradigmas e ser capaz de trabalhar em colaboração com outros. |
| Évaluation | Avaliação |
| Vous pouvez faire cette évaluation individuellement ou en groupe avant de tenter de remplir correctement les évaluations formatives. | |
| Cette évaluation est sur l'activité vient de s'achever. | Essa avaliação é baseada na atividade que acabou de ser concluída. |
| Lors de votre activité de recherche, quel est le langage qui est commun et le plus préféré par les développeurs? | Durante sua atividade de pesquisa, qual é a linguagem comum e a mais preferida pelos desenvolvedores? |
| Quelles sont quelques raisons de la préférence du langage mentionné (à la question 1 ci-dessus). | Quais são algumas das razões para a preferência de linguagem mencionada (na questão 1 acima). |
| Comment les programmeurs choisissent le langage à utiliser pour résoudre un problème particulier. | Como os programadores escolhem qual linguagem usar para resolver um problema específico. |
| Résumé de l'unité | Resumo da unidade |
| Cette unité a permis d'apprendre ce qu'est la programmation, l'histoire de la programmation, les | Esta unidade permitiu aprender o que é programação, a história da programação, os níveis, |

| | |
|--|--|
| niveaux, les générations et les paradigmes des langages de programmation. | as gerações e os paradigmas das linguagens de programação. |
| Jusqu'ici, l'apprenant comprend les concepts de base de la programmation et est prêt à passer à l'unité suivante. | Até agora, o aluno entende os conceitos básicos de programação e está pronto para passar para a próxima unidade. |
| Évaluation de l'unité | Avaliação da unidade |
| Système de notation | Sistema de classificação |
| Question 1 : | Pergunta 1: |
| 5 points | 5 pontos |
| Question 2 : | Pergunta 2: |
| 2 points | 2 pontos |
| Question 3 : | Pergunta 3: |
| 1 point | 1 ponto |
| Question 4 : | Pergunta 4: |
| 2 points | |
| Question 5 : | Pergunta 5: |
| 1 point | |
| Instructions | Instruções |
| Le but de cette évaluation est de vérifier les progrès de l'apprenant en déterminant ce que l'apprenant a appris dans cette unité. | O objetivo desta avaliação é verificar o progresso do aluno, determinando o que o aluno aprendeu nesta unidade. |
| Les questions couvrent tout ce qui a été présenté dans cette unité afin d'évaluer la compréhension globale. | As perguntas abrangem tudo o que foi apresentado nesta unidade, a fim de avaliar a compreensão geral. |
| Répondez attentivement et si votre score tombe en dessous de 40%, refaire les lectures. | Responda com cuidado e se sua pontuação cair abaixo de 40%, repita as leituras. |
| entre 40% et 60%, refaire les lectures sur votre zone de faiblesse | entre 40% e 60%, repita as leituras na sua zona de dificuldade |
| supérieur à 60%, vous avez une quantité importante de connaissances | maior que 60%, você tem uma quantidade significativa de conhecimento |
| Questions | Perguntas |
| La programmation peut être définie comme le fait d'utiliser un pour écrire des | A programação pode ser definida como usar um para escrever |
| qui peuvent être exécutées par l'ordinateur dans le but de en appliquant les | que pode ser executado pelo computador com a finalidade de aplicar |
| et correctes du langage de programmation. | e linguagem de programação correta. |
| a été la première génération de langage | foi a primeira geração de linguagem |
| d'ordinateur et utilise le langage pour écrire les instructions | computador e usa o idioma para escrever instruções |
| Lequel des langages suivants utilise des mnémoniques pour représenter les instructions | Qual das seguintes linguagens usa mnemônicos para representar instruções |
| a. | a. |
| Langage de haut niveau b. langage de bas niveau c. langage assemblage | Linguagem de alto nível b. Linguagem de baixo nível c. Linguagem Assembly |
| Un langage de haut niveau utilise | Usos de linguagem de alto nível |
| ou | ou |
| pour traduire les instructions en langage machine. | para traduzir instruções para a linguagem de máquina. |
| Lesquels des langages suivants supportent le paradigme orienté objet | Quais das seguintes linguagens suportam o paradigma orientado a objetos |

| | |
|--|--|
| A. | A. |
| C B. C ++ C. Pascal | C B. C ++ C. Pascal |
| D. Java | D. Java |
| E. Visual Basic.Net | E. Visual Basic.Net |
| Réponses | Respostas |
| langage de programmation, instructions, résoudre un problème, les syntaxes, la sémantique | linguagem de programação, instruções, resolução de problemas, sintaxes, semântica |
| Le langage machine, bas niveau | Linguagem de máquina, baixo nível |
| Assemblage, | Montagem |
| des compilateurs ou des interpréteurs | compiladores ou intérpretes |
| C++, Java, Visual Basic.Net | C++, Java, Visual Basic.Net |
| Unité 1. | |
| Introduction à la programmation | |
| Unité 1. | |
| Introduction à la programmation | |
| Unité 2. | Unidade 2. |
| Résolution de problème basée sur l'informatique | Solução de problemas baseada em computador |
| <i>"The sooner you start coding your program, the longer it is going to take" [H.F. ledgard, programming proverb]</i> | <i>"Quanto mais cedo você começar a codificar seu programa, mais tempo vai demorar" [H.F. Ledgard, Provérbio de Programação]</i> |
| Introduction de l'unité | Introdução da unidade |
| Dans l'unité précédente, nous avons défini la programmation comme le moyen d'utiliser les ordinateurs pour résoudre les problèmes avec le seul but de rendre le travail facile et augmenter la productivité. | Na unidade anterior, definimos a programação como a forma de usar computadores para resolver problemas com o único propósito de facilitar o trabalho e aumentar a produtividade. |
| Toujours dans l'activité 1, nous étions chargés d'identifier quelques problèmes qui peuvent être résolus à l'aide d'un ordinateur. | Também na Atividade 1, fomos encarregados de identificar alguns problemas que podem ser resolvidos usando um computador. |
| Cette unité vous aidera à découvrir si vous les avez bien fait ou non. | Esta unidade o ajudará a descobrir se você os fez bem ou não. |
| Ainsi la programmation implique l'automatisation des systèmes, des processus, des opérations ou une activité. | Portanto, a programação envolve a automação de sistemas, processos, operações ou uma atividade. |
| La solution (programme) doit satisfaire les besoins ou les exigences de l'utilisateur final. | A solução (programa) deve satisfazer as necessidades ou requisitos do usuário final. |
| L'utilisateur final peut être un agriculteur dans le village qui a besoin des mises à jour quotidiennes sur les prix du marché, y compris des informations qui peuvent aider à améliorer la production. | O usuário final pode ser um agricultor na aldeia que precisa de atualizações diárias sobre os preços de mercado, incluindo informações que podem ajudar a melhorar a produção. |
| Par conséquent, pour une solution à réaliser, les programmeurs doivent d'abord comprendre le problème et suivre les processus de développement de logiciels qui entraînent, l'identification des problèmes, l'analyse, la conception de la solution qui peut être réalisé grâce à l'utilisation des techniques de conception descendante (top down) ou ascendante (bottom up). | Portanto, para que uma solução seja alcançada, os programadores devem primeiro entender o problema e seguir os processos de desenvolvimento de software resultantes, identificação de problemas, análise, design de solução que podem ser alcançados através do uso de técnicas de design top-down ou bottom-up. |

| | |
|--|---|
| Les solutions peuvent également être conçues en utilisant des organigrammes (flowchart) et des algorithmes et / ou pseudocodes. | As soluções também podem ser projetadas usando fluxogramas e algoritmos e/ou pseudocódios. |
| Par conséquent, cette unité se focalise sur l'analyse des problèmes et les techniques de conception de solutions à un niveau très basique. | Portanto, esta unidade se concentra na análise de problemas e técnicas de design de soluções em um nível muito básico. |
| Objectifs de l'unité | Objetivos da unidade |
| À la fin de cette unité, vous devriez être capable de : | |
| Formuler et définir les problèmes. | Formule e defina problemas. |
| Concevoir des solutions en appliquant les techniques de conception. | Soluções de design aplicando técnicas de design. |
| Concevoir des organigrammes (flowchart). | Fluxogramas de design. |
| Décrire les règles de développement d'un organigramme. | Descreva as regras para o desenvolvimento de um organograma. |
| Représenter des solutions en utilisant des algorithmes et pseudocodes. | Representar soluções usando algoritmos e pseudocódios. |
| Décrire les caractéristiques d'un algorithme et d'un pseudocode. | Descreva as características de um algoritmo e de um pseudocódigo. |
| Décrire les techniques de programmation. | Descreva técnicas de programação. |
| Principes de Programmation | |
| Principes de Programmation | |
| Unité 2. | Unidade 2. |
| Résolution de problème basée sur l'informatique | Solução de problemas baseada em computador |
| Unité 2. | |
| Résolution de problème basée sur l'informatique | |
| Termes clés | |
| Algorithme : | Algoritmo: |
| Un algorithme est une procédure consistant en un ensemble fini de règles claires (instructions) qui spécifient une séquence finie d'opérations qui fournit la solution à un problème ou à une classe spécifique de problèmes pour tout ensemble admissible de variables d'entrée (s'il y a des entrées). | Um algoritmo é um procedimento que consiste em um conjunto finito de regras claras (instruções) que especificam uma sequência finita de operações que fornece a solução para um problema específico ou classe de problemas para qualquer conjunto permitido de variáveis de entrada (se houver entradas). |
| En d'autres termes, un algorithme est une procédure d'étape par étape pour résoudre un problème donné. | Em outras palavras, um algoritmo é um procedimento passo a passo para resolver um determinado problema. |
| Analyse ascendante : | Análise crescente: |
| C'est une technique de conception où un programme est divisé en un certain nombre de tâches plus petites et plus simples qui peuvent être abordées séparément. | É uma técnica de design em que um programa é dividido em várias tarefas menores e mais simples que podem ser abordadas separadamente. |
| Dans cette technique, des sous programmes sont écrits et testés en premier avant que le programme principal ne soit écrit. | Nesta técnica, os subprogramas são escritos e testados primeiro antes que o programa principal seja escrito. |
| Design : le design est le développement des alternatives du problème et des modèles pour représenter le problème et la solution. | Design: Design é o desenvolvimento de alternativas ao problema e modelos para representar o problema e a solução. |

| | |
|---|---|
| C'est un processus clair à travers lequel les besoins des utilisateurs sont transformés en une représentation compréhensible par la machine. | É um processo claro através do qual as necessidades dos usuários são transformadas em uma representação que pode ser entendida pela máquina. |
| Implémentation : | Implantação: |
| Il s'agit du codage et du test du programme | Trata-se de codificar e testar o programa |
| Organigramme : | Organograma: |
| C'est une technique normalisée pour représenter graphiquement des schémas avec un ensemble de symboles normalisés qui représentent la logique du programme. | É uma técnica padronizada para representar graficamente esquemas com um conjunto de símbolos padronizados que representam a lógica do programa. |
| Modèle : est une abstraction de la réalité. | Modelo: é uma abstração da realidade. |
| Modularisation : entraîne le morcellement d'un vaste programme en sous-programmes qui sont gérables de par la taille en termes de complexité de la logique et du nombre d'instructions. | Modularização: faz com que um grande programa seja dividido em sub-rotinas que são gerenciáveis por tamanho em termos de complexidade da lógica e número de instruções. |
| Module : est une partie d'un programme qui remplit une fonction distincte | Módulo: é uma parte de um programa que executa uma função separada |
| Planning : c'est le niveau de base du développement d'un logiciel où le problème est défini et priorisé. | Planejamento: este é o nível básico de desenvolvimento de software em que o problema é definido e priorizado. |
| Programmation structurée : c'est un paradigme de programmation qui supporte la modularité en morcelant les fonctions dans des modules triviaux. | Programação estruturada: é um paradigma de programação que suporta a modularidade dividindo funções em módulos triviais. |
| Principes de Programmation | |
| Principes de Programmation | |
| Déploiement : le déploiement entraîne l'installation du programme dans l'environnement de l'utilisateur et la formation des utilisateurs. | Implantação: a implantação instala o programa no ambiente do usuário e treina os usuários. |
| Chemin logique : | Caminho lógico: |
| C'est une séquence d'instructions ou déclarations également connue sous le nom d'algorithme. | É uma sequência de instruções ou declarações também conhecida como algoritmo. |
| Maintenance : implique d'effectuer des changements sur le programme après l'installation. | Manutenção: envolve fazer alterações no programa após a instalação. |
| Problème : est une situation indésirable qui empêche les utilisateurs de réaliser pleinement leurs objectifs. | Problema: é uma situação indesejada que impede que os usuários atinjam plenamente seus objetivos. |
| Un problème et la nécessité de le résoudre découle d'une volonté de transformer la situation actuelle en une autre situation plus désirée (Programming and PROB solving using C par ISRD). | Um problema e a necessidade de resolvê-lo decorre do desejo de transformar a situação atual em outra situação mais desejada (Programação e resolução de PROB usando C por ISRD). |
| Processus de Développement de Programme : est un ensemble normalisé de mesures utilisées pour fournir une approche logique, le bon sens (ou processus) pour résoudre un problème difficile avec une application informatique. | Processo de Desenvolvimento de Programas: é um conjunto padronizado de medidas usadas para fornecer uma abordagem lógica, de bom senso (ou processo) para resolver um problema difícil com um aplicativo de computador. |
| Pseudocode : est une manière générique de décrire un algorithme sans une utilisation de toute syntaxe spécifique à un langage de programmation. | Pseudocódigo: é uma forma genérica de descrever um algoritmo sem o uso de qualquer sintaxe específica para uma linguagem de programação. |

| | |
|---|--|
| C'est la représentation d'une solution en utilisant l'anglais non standard comme un langage de programmation. | É a representação de uma solução usando inglês não padrão como linguagem de programação. |
| Il utilise des déclarations/mots et expressions de l'anglais courant que ce qui est généralement trouvé dans la syntaxe des langages de programmation. | Ele usa declarações/palavras e expressões do inglês fluente do que o que geralmente é encontrado na sintaxe das linguagens de programação. |
| Logique de programme : est utilisée par le programmeur pour modéliser les instructions du langage de programmation effectuées par l'ordinateur lorsque le programme est exécuté. | Lógica do programa: é usada pelo programador para modelar instruções de linguagem de programação executadas pelo computador quando o programa é executado. |
| Déclarations du Langage de programmation : ce sont des déclarations basées sur l'anglais qui, lorsqu'elles sont exécutées dans la séquence correcte peut demander à l'ordinateur d'effectuer une série de tâches. | Declarações de linguagem de programação: são instruções baseadas em inglês que, quando executadas na sequência correta, podem instruir o computador a executar uma série de tarefas. |
| Elles sont utilisées pour mettre en œuvre une logique de programme par l'envoi d'instructions au système d'exploitation. | Eles são usados para implementar a lógica do programa enviando instruções para o sistema operacional. |
| Analyse des besoins : | Análise das necessidades: |
| C'est la deuxième phase de développement d'un logiciel après la planification. | Esta é a segunda fase do desenvolvimento de software após o planejamento. |
| Il s'agit de rechercher les spécifications du problème par le biais de questionnaires, des entretiens et d'observations entre autres méthodes. | Envolve pesquisar as especificações do problema por meio de questionários, entrevistas e observações, entre outros métodos. |
| Cycle de Vie de Développement d'un Logiciel : | Ciclo de vida de desenvolvimento de software: |
| C'est la méthodologie ou processus utilisé par les personnes qui travaillent dans la technologie de l'information pour résoudre des problèmes en fournissant une solution efficace et efficiente. | É a metodologia ou processo usado por pessoas que trabalham com tecnologia da informação para resolver problemas, fornecendo uma solução eficaz e eficiente. |
| Analyse descendante : | Análise de cima para baixo: |
| C'est une technique de conception où un programme est divisé en un certain nombre de tâches plus petites et plus simples qui peuvent être abordés séparément. | É uma técnica de design em que um programa é dividido em várias tarefas menores e mais simples que podem ser abordadas separadamente. |
| Dans cette technique, le programme principal est écrit et testé avant que les sous-programmes ne soient écrits. | Nesta técnica, o programa principal é escrito e testado antes que as sub-rotinas sejam escritas. |
| Ainsi, il fait voir en premier la solution dans un cadre plus large. | Assim, ele mostra a solução primeiro em uma estrutura mais ampla. |
| Activités d'apprentissage | Atividades de aprendizagem |
| Activité 2.1 - Résolution de problèmes | Atividade 2.1 - Resolução de problemas |
| Introduction | Introdução |
| Différentes étapes sont suivies afin d'arriver à une solution ou développer un logiciel. | Diferentes etapas são seguidas para chegar a uma solução ou desenvolver um software. |
| Le premier niveau de résolution de problèmes comprend la définition du problème qui implique la compréhension de la nature du problème à portée de main. | O primeiro nível de resolução de problemas inclui a definição do problema, que envolve a compreensão da natureza do problema em questão. |

| | |
|--|---|
| Ce niveau de résolution de problèmes n'aborde pas comment résoudre le problème, mais nous dit ce que le problème est vraiment. | Esse nível de resolução de problemas não aborda como resolver o problema, mas nos diz qual é realmente o problema. |
| Il s'agit de comprendre les entrées, les opérations et la sortie attendue du programme. | Trata-se de entender as entradas, operações e saída esperada do programa. |
| Le schéma suivant illustre les étapes suivies lors de la résolution d'un problème. | O diagrama a seguir ilustra as etapas que são tomadas ao resolver um problema. |
| Unité 2. | |
| Résolution de problème basée sur l'informatique | |
| Unité 2. | |
| Résolution de problème basée sur l'informatique | |
| Principes de Programmation | |
| Principes de Programmation | |
| Figure 2.1 : | Figura 2.1: |
| Les étapes de résolution de problèmes (source : Programming and PROB solving using C écrit par ISRD) | Etapas de resolução de problemas (fonte: Programação e resolução de PROB usando C (escrito por ISRD)) |
| Prendre connaissance du problème | Saiba mais sobre o problema |
| Recueillir les informations nécessaires pour accroître la compréhension du problème | Reúna as informações necessárias para aumentar a compreensão do problema |
| Développer des solutions alternatives | Desenvolva soluções alternativas |
| Développer des critères d'évaluation | Desenvolver critérios de avaliação |
| Choisir la solution la plus appropriée | Escolhendo a solução mais adequada |
| Implémenter la solution | Implementar a solução |
| Surveiller et contrôler l'implémentation | Monitore e controle a implementação |
| Définir le problème en termes d'objectifs, de paramètres et leurs interactions | Definir o problema em termos de objetivos, parâmetros e suas interações |
| "Il n'y a pas de livres de cuisine sur les stratégies pour remplacer l'intelligence, l'expérience et le bon goût de la programmation", Harsha Priya, R. Ranjeet. | "Não há livros de receitas sobre estratégias para substituir inteligência, experiência e bom gosto em programação", Harsha Priya, R. Ranjeet. |
| Détails de l'activité | Detalhes da atividade |
| Par conséquent, dans cette activité, vous devez lire les sujets sur ce que sont vraiment les problèmes de programmation, pourquoi les ordinateurs et autres systèmes numériques sont utilisés pour résoudre les problèmes, la définition de problème, les similitudes entre les problèmes, les stratégies de résolution de problèmes et les étapes de résolution d'un problème informatique. | Portanto, nesta atividade, você deve ler os tópicos sobre o que realmente são problemas de programação, por que computadores e outros sistemas digitais são usados para resolver problemas, definição de problemas, semelhanças entre problemas, estratégias de resolução de problemas e etapas para resolver um problema. problema de computador. |
| Les lectures peuvent être obtenues à partir de (Programming and Prob solving using C écrit par ISRD, Programming and problem Solving Through C Language écrit par Harsha Priya, R. Ranjeet, et Problem solving and program design in C écrit par Jeri R. Hanly & Elliot B. Koffman) et de la liste des livres fournis dans la section référence pour cette unité. | As leituras podem ser obtidas em (Programação e resolução de problemas usando C escrito por ISRD, Programação e resolução de problemas através da linguagem C escrito por Harsha Priya, R. Ranjeet, e Resolução de problemas e design de programa em C escrito por Jeri R. Hanly & Elliot B. Koffman) e a lista de livros fornecidos na referência seção para esta unidade. |

| | |
|--|--|
| Les apprenants devraient également compléter les lectures en faisant des recherches individuelles afin d'améliorer leur compréhension. | |
| Dans cette activité, vous êtes tenus de rédiger des notes brèves sur : | Nesta atividade, você deve escrever breves notas sobre: |
| Pourquoi un ordinateur est-il un outil nécessaire pour la résolution des problèmes. | Por que um computador é uma ferramenta necessária para solucionar problemas. |
| Les stratégies de résolution de problème et leurs avantages dans la programmation. | Estratégias de resolução de problemas e suas vantagens na programação. |
| Processus de développement de logiciel. | Processo de desenvolvimento de software. |
| Similitudes entre les problèmes qui peuvent être résolus en utilisant un ordinateur. | Semelhanças entre problemas que podem ser resolvidos usando um computador. |
| Conclusion | Conclusão |
| Cette activité est destinée à aider l'apprenant à acquérir une meilleure compréhension de ce que sont les problèmes en programmation, connaître les stratégies et les étapes de la conception d'une solution et les similarités entre les problèmes. | Esta atividade tem como objetivo ajudar o aluno a entender melhor quais são os problemas na programação, as estratégias e etapas envolvidas na concepção de uma solução e as semelhanças entre os problemas. |
| Évaluation | |
| Qu'est-ce un problème? | O que é um problema? |
| Qu'entendez-vous par le terme définition de problème? | O que você quer dizer com o termo definição de problema? |
| Ordinateurs et autres appareils numériques tels que les téléphones mobiles sont utilisés pour résoudre des problèmes. | Computadores e outros dispositivos digitais, como telefones celulares, são usados para resolver problemas. |
| Donnez les raisons pour lesquelles ces dispositifs sont utilisés comme outils de résolution de problèmes. | Indique os motivos pelos quais esses dispositivos são usados como ferramentas de solução de problemas. |
| Pourquoi devriez-vous appliquer des stratégies dans la résolution de problèmes? et quelles sont ces stratégies? | Por que você deve aplicar estratégias na resolução de problemas? e quais são essas estratégias? |
| Discute Le développement d'un programme entraîne quelques étapes et l'une des étapes implique la modélisation de la logique. | Discute O desenvolvimento de um programa envolve algumas etapas e uma das etapas envolve a lógica de modelagem. |
| Cependant, il est possible d'arriver à une solution sans adhérer à ces étapes. | No entanto, é possível chegar a uma solução sem seguir essas etapas. |
| Pourquoi est-il nécessaire d'appliquer le processus de développement des logiciels? | Por que é necessário aplicar o processo de desenvolvimento de software? |
| Expliquez | Explique |
| La définition du problème transmet les entrées, les sorties et les opérations du programme. | A definição do problema transmite as entradas, saídas e operações do programa. |
| Discutez. | Discutir. |
| Les similitudes entre les problèmes peuvent inclure la vitesse de traitement, le volume de travail, la fiabilité, la précision, la sécurité, le coût, la recherche et les manipulations des chaînes de caractères. | As semelhanças entre os problemas podem incluir velocidade de processamento, volume de trabalho, confiabilidade, precisão, segurança, custo, pesquisa e manipulações de cadeia. |
| Discutez de ces similitudes. | Discuta essas semelhanças. |
| Activité 2.2 - Conception de programme | Atividade 2.2 - Design do programa |
| Introduction | Introdução |

| | |
|---|--|
| La conception du programme implique comment résoudre le problème et est une phase de résolution de problème qui vient après la définition du problème. | O design do programa envolve como resolver o problema e é uma fase de resolução de problemas que vem depois que o problema é definido. |
| Il s'agit de fournir des solutions aux problèmes en appliquant différentes techniques de conception, y compris la modélisation. | Trata-se de fornecer soluções para problemas aplicando diferentes técnicas de design, incluindo modelagem. |
| Ces techniques incluent le morcellement d'un programme en des sous-programmes. | Essas técnicas incluem dividir um programa em subprogramas. |
| Cet aspect de diviser un programme en sous-programmes conduit à la modularisation. | Esse aspecto de dividir um programa em subprogramas leva à modularização. |
| La modularisation d'autre part, est l'un des éléments de base de la programmation structurée. | A modularização, por outro lado, é um dos elementos básicos da programação estruturada. |
| Algorithmes | Algoritmos |
| Les algorithmes sont un ensemble d'étapes générées par le programmeur pour l'aider à résoudre un problème particulier. | Algoritmos são um conjunto de etapas geradas pelo programador para ajudá-lo a resolver um problema específico. |
| Parce qu'un algorithme est utilisé comme un outil de conception, il doit donc être fini, sans ambiguïté, simple (définitive) et précise, complète, efficace et permettre l'entrée et la sortie des données. | Como um algoritmo é usado como uma ferramenta de design, ele deve, portanto, ser finito, inequívoco, simples (definitivo) e preciso, completo, eficiente e permitir a entrada e saída de dados. |
| Les algorithmes sont indépendants du langage de programmation et peuvent être développés en utilisant trois grandes constructions de programmation qui comprennent : | Os algoritmos são independentes da linguagem de programação e podem ser desenvolvidos usando três grandes construções de programação que incluem: |
| La construction séquentielle où les étapes sont écrites dans un ordre particulier, par exemple d'abord, l'étape A est exécutée puis B et C vient après B etc. Le flux de contrôle se fait étape par étape. | A construção sequencial em que as etapas são escritas em uma ordem específica, por exemplo primeiro, a etapa A é executada, depois B e C vem depois de B etc. O fluxo de controle é feito passo a passo. |
| Par exemple : | Por exemplo: |
| Lire a, b | Leia a, b |
| sum= a + b | soma = a + b |
| Afficher sum | Mostrar soma |
| La deuxième construction est conditionnelle. | A segunda construção é condicional. |
| Cela signifie qu'à un certain point dans l'exécution des étapes, et pour réaliser une tâche, il est nécessaire de se brancher ou de prendre une décision en évaluant une condition. | Isso significa que, em algum momento da execução das etapas, e para concluir uma tarefa, é necessário conectar ou tomar uma decisão avaliando uma condição. |
| Par exemple : | |
| Lire a, b | |
| sum = a + b | soma = a + b |
| si (sum > d) | se (soma > d) |
| alors Afficher sum | em seguida, Mostrar soma |
| Sinon | Caso contrário |
| Afficher d | Exibir d |
| L'autre construction est la boucle. | A outra construção é o loop. |

| | |
|--|---|
| Dans cette construction, une étape est répétée un nombre de fois avant l'exécution de l'étape suivante ou d'autres étapes en fonction d'une condition. | Nesta construção, uma etapa é repetida várias vezes antes de executar a próxima etapa ou outras etapas, dependendo de uma condição. |
| Lorsque la condition devient fautive, la boucle est quittée et d'autres étapes sont alors exécutées. | Quando a condição se torna falsa, o loop é encerrado e outras etapas são executadas. |
| Par exemple: | Por exemplo: |
| Lire a, b | |
| sum = a + b | |
| Tant que (a<b) | Contanto que (a<b) |
| Afficher a | Mostrar um |
| Lire next a, b | Leia a seguir a, b |
| répéter l'évaluation | repita a avaliação |
| Rappelez-vous, nous avons défini un algorithme comme un bel ensemble d'instructions qui signifie qu'il a un début et une fin. | Lembre-se, definimos um algoritmo como um bom conjunto de instruções, o que significa que ele tem um começo e um fim. |
| Un exemple d'algorithme qui additionne deux nombres. | Um exemplo de um algoritmo que adiciona dois números juntos. |
| début | iniciar |
| valeurs d'entrée pour A et B | valores de entrada para A e B |
| ajouter a à b et le stocker dans total | adicione a a b e guarde-o no total |
| afficher la valeur de la somme | exibir o valor da soma |
| fin | final |
| Organigrammes | Organogramas |
| L'organigramme est une représentation schématique d'un algorithme. | O fluxograma é uma representação esquemática de um algoritmo. |
| Un organigramme est conçu en utilisant les symboles suivants : | Um organograma é projetado usando os seguintes símbolos: |
| Afficher total | Mostrar total |
| Total = | Total = |
| a + b | um + b |
| Lire a et b | Leia a e b |
| Figure 2.2 : | Figura 2.2: |
| Exemple d'organigramme | Exemplo de um organograma |
| Fin | Termina |
| Début | Iniciar |
| Maintenant, nous pouvons représenter notre algorithme (ci-dessus) en utilisant un organigramme. | Agora, podemos representar nosso algoritmo (acima) usando um fluxograma. |
| Bouclet | Bouclet |
| Entrée/sortie | Entrada/saída |
| Connecteur | conectora |
| Décision | Decisão |
| Traitement | Tratamento |
| Début/fin | Início/fim |
| 39 | 39 |
| Technique de conception descendante (top-down) | Técnica de design de cima para baixo |
| Cette stratégie de conception élimine la complexité de gestion d'un gros problème en les décomposant en sous-modules pouvant être facilement compris. | Essa estratégia de design elimina a complexidade de gerenciar um grande problema, dividindo-os |

| | |
|---|---|
| | em submódulos que podem ser facilmente compreendidos. |
| Le processus de division d'une tâche en sous-tâche est répété jusqu'à ce qu'on atteigne une tâche facile à mettre en œuvre. | O processo de dividir uma tarefa em uma subtarefa é repetido até que uma tarefa fácil de implementar seja alcançada. |
| La figure ci-dessous montre comment la technique descendante peut être utilisée. | A figura abaixo mostra como a técnica de cima para baixo pode ser usada. |
| Sous fonction 3 | Sub função 3 |
| Fonctions de sortie | Funções de saída |
| PROGRAMME PRINCIPAL | PROGRAMA PRINCIPAL |
| Fonctions d'entrée | Funções de entrada |
| Fonctions principales | Funções principais |
| Sous fonction 2 | Sub função 2 |
| Sous fonction 1 | Sub função 1 |
| Figure 2.3 : | Figura 2.3: |
| Technique de conception top-down (source : | Técnica de design de cima para baixo (fonte: |
| Harsha | Harsha |
| Priya, R. Ranjeet) | Priya, R. Ranjeet) |
| Avantages | Benefícios |
| Le principal avantage de l'approche top-down est qu'il permet dans un premier temps de voir le programme dans sa globalité. | A principal vantagem da abordagem de cima para baixo é que ela permite que você veja o programa como um todo no início. |
| Pour écrire les modules de haut niveau, on n'a pas besoin de se soucier des petits détails. | Para escrever módulos de alto nível, você não precisa se preocupar com pequenos detalhes. |
| Les modules de niveau inférieur sont conçus en ayant à l'esprit les modules de niveau supérieur et on sait déjà dans les moindres détails, les objectifs à atteindre. | Os módulos de nível inferior são projetados com os módulos de nível superior em mente e já sabemos detalhadamente os objetivos a serem alcançados. |
| Ce qui permet de garantir la compatibilité entre des niveaux élevés et les niveaux inférieurs. | Isso garante a compatibilidade entre níveis mais altos e mais baixos. |
| L'approche top-down convient aux problèmes de grande taille. | A abordagem de cima para baixo é adequada para grandes problemas. |
| Inconvénients | Contras |
| La solution offre une couverture limitée dans les premières phases | A solução oferece cobertura limitada nos estágios iniciais |
| On ne se rend pas vite compte des avancés du projet. | Não percebemos rapidamente o andamento do projeto. |
| Elle engendre la création d'outils particuliers et différents même si les problèmes traités sont similaires dans les différentes branches. | Ele gera a criação de ferramentas particulares e diferentes, mesmo que os problemas tratados sejam semelhantes nos diferentes ramos. |
| etc... | etc... |
| <i>Lire programming and problem solving through c language écrit par Harsha Priya, R. Ranjeet, et également, Programming and PROB solving using C écrit par ISRD pour comprendre les avantages et les inconvénients de la technique de conception top-down.</i> | <i>Leia Programação e Resolução de Problemas Através da Linguagem C escrita por Harsha Priya, R. Ranjeet, e também Programação e Resolução de PROB Usando C escrito por ISRD para entender os prós e contras da técnica de design top-down.</i> |
| Unité 2. | |
| Résolution de problème basée sur l'informatique | |
| Unité 2. | |

| | |
|--|---|
| Résolution de problème basée sur l'informatique | |
| L'approche bottom-up améliore la lisibilité du code. | A abordagem de baixo para cima melhora a legibilidade do código. |
| Permet la réutilisation du code. | Permite que o código seja reutilizado. |
| L'approche bottom-up est aussi simple car il n'est pas nécessaire de créer des talons pour les modules de niveau supérieur. | A abordagem de baixo para cima também é simples porque não há necessidade de criar saltos para módulos de nível superior. |
| En effet, les talons sont écrits et testés avant de progresser. | Na verdade, os saltos são escritos e testados antes de progredir. |
| Facilite le prototypage. | Facilita a prototipagem. |
| L'approche bottom-up est mieux adaptée aux petits problèmes. | A abordagem de baixo para cima é mais adequada para pequenos problemas. |
| etc... | |
| Inconvénients | Contras |
| On n'a pas une vue globale du programme au premier abord. | Não temos uma visão global do programa à primeira vista. |
| Lorsqu'on écrit les modules de niveau inférieur, on n'a pas à l'esprit ceux de niveau supérieur. | Ao escrever módulos de nível inferior, os módulos de nível superior não estão em mente. |
| Cela peut conduire à une mauvaise connexion entre les modules de niveau supérieur et ceux de niveau inférieur lors de l'intégration des différents modules. | Isso pode levar a uma conexão ruim entre os módulos de nível superior e inferior ao integrar os diferentes módulos. |
| Cette approche rend plus difficile la surveillance et la gestion du projet. | Essa abordagem dificulta o monitoramento e o gerenciamento do projeto. |
| <i>Lire programming and problem solving through c language écrit par Harsha Priya, R. Ranjeet, et également, Programming and PROB solving using C écrit par ISRD pour comprendre les avantages et les inconvénients de la technique de conception bottom-up.</i> | <i>Leia Programação e Resolução de Problemas Através da Linguagem C escrita por Harsha Priya, R. Ranjeet, e também Programação e Resolução de PROB Usando C escrito por ISRD para entender os prós e contras da técnica de design de baixo para cima.</i> |
| Technique de conception ascendante (bottom-up) | Técnica de design de baixo para cima |
| Ici, les sous-programmes les plus élémentaires sont écrits et plus tard utilisés pour faire des sous-routines sophistiquées. | Aqui, as sub-rotinas mais básicas são escritas e posteriormente usadas para criar sub-rotinas sofisticadas. |
| Pourquoi cette approche n'est pas très recommandée ? | Por que essa abordagem não é altamente recomendada? |
| C'est comme l'inversion de la technique de conception descendante, voir la figure ci-dessous. | É como inverter a técnica de design de cima para baixo, veja a figura abaixo. |
| Sous fonction 3 | |
| Sous fonction 2 | |
| Sous fonction 1 | Sub função 1 |
| Fonctions de sortie | |
| Fonctions principales | |
| Fonctions d'entrée | |
| PROGRAMME PRINCIPAL | |
| Figure 2.4 : | Figura 2.4: |
| Technique de conception bottom-up (source : | Técnica de design de baixo para cima (fonte: |
| Harsha Priya, R. Ranjeet) | Harsha Priya, R. Ranjeet) |
| Avantages | Benefícios |

| | |
|--|---|
| Vue d'ensemble des techniques de conception de la programmation | Visão geral das técnicas de design de programação |
| Programmation Linéaire : | Programação linear: |
| Cela nécessite une programmation de manière séquentielle. | Isso requer programação de forma sequencial. |
| La prise de décision et constructions d'itérations ne sont pas inclus. | A tomada de decisão e as compilações de iteração não estão incluídas. |
| C'est une traduction directe d'un algorithme séquentiel. | É uma tradução direta de um algoritmo sequencial. |
| Programmation structurée : | Programação estruturada: |
| Cette technique de conception de la programmation utilise trois grandes constructions de programmation : séquentielle, les boucles et les structures conditionnelles. | Essa técnica de design de programação usa três construções de programação principais: estruturas sequenciais, loops e condicionais. |
| Il englobe également l'utilisation des techniques de résolution de problèmes top-down (expliqué ci-dessus). | Também abrange o uso de técnicas de resolução de problemas de cima para baixo (explicado acima). |
| Par conséquent, les programmes écrits en utilisant cette approche peuvent être facilement compris, corrigés (débogués), réutilisables, etc. Lire Programming and PROB solving using C écrit par ISRD pour comprendre les avantages et les éléments de la programmation structurée. | Portanto, os programas escritos usando essa abordagem podem ser facilmente compreendidos, corrigidos (depurados), reutilizáveis etc. leitura de programação e resolução de PROB usando C escrito por ISRD para entender os benefícios e elementos da programação estruturada. |
| La conception modulaire des programmes : | Projeto de programa modular: |
| Les techniques descendantes conduisent à la notion de modularisation. | Técnicas de cima para baixo levam à noção de modularização. |
| Les programmes peuvent être logiquement séparés en : l'initialisation, l'entrée, la validation des données d'entrée, le traitement, la sortie, la gestion des erreurs et la procédure de fermeture. | Os programas podem ser logicamente separados em: inicialização, entrada, validação de dados de entrada, processamento, saída, tratamento de erros e procedimento de desligamento. |
| Lire Programming and PROB solving using C écrit par ISRD et comprendre les attributs de base d'un module, de contrôler les relations entre les modules, la communication entre les modules et les exigences de conception du module. | Leia Programming for Problem Solving (Using C) por ISRD e entenda os atributos básicos de um módulo, relações de controle entre módulos, comunicação entre módulos e requisitos de design de módulo. |
| Activité pratique | Atividade prática |
| En utilisant un crayon and du papier ou un ordinateur, construire un organigramme pour l'un des deux problèmes listés dans l'activité 1. | Usando papel e caneta ou um computador, construa um fluxograma para um dos dois problemas listados na Atividade 1. |
| Ecrire un algorithme pour l'organigramme que vous avez construit ci-dessus | Escreva um algoritmo para o fluxograma que você criou acima |
| Testez votre algorithme utilisant la technique de vérification desk checking. | Teste seu algoritmo usando a técnica de verificação de mesa. |
| Détails de l'activité | Detalhes da atividade |
| Ainsi, vous devez lire des sujets sur les techniques de conception qui comprennent les techniques de conception ascendante et descendante; leurs avantages et leurs inconvénients et où chaque technique est plus applicable. | Portanto, você precisa ler tópicos sobre técnicas de design que incluem técnicas de design de baixo para cima e de cima para baixo; seus prós e contras e onde cada técnica é mais aplicável. |

| | |
|--|--|
| En outre, lire sur les algorithmes; les caractéristiques d'un bon algorithme, la logique de l'algorithme (flux séquentielle, conditionnelle et répétitive), la façon de tester un algorithme en utilisant la technique de vérification appelée desk checking et la technique appelée walkthrough, la différence entre un algorithme et un pseudo-code, les avantages de l'utilisation d'un pseudo-code sur les organigrammes et vice versa et la construction d'organigramme; y compris les avantages et les inconvénients de l'utilisation des organigrammes, les règles de conception d'un organigramme. | Além disso, leia sobre algoritmos; as características de um bom algoritmo, a lógica do algoritmo (fluxo sequencial, condicional e repetitivo), como testar um algoritmo usando a técnica de verificação chamada verificação de mesa e a técnica chamada passo a passo, a diferença entre um algoritmo e um pseudo-código, as vantagens do uso de pseudo-código em fluxogramas e vice-versa e construção de fluxogramas; incluindo prós e contras do uso de fluxogramas, regras para projetar um organograma. |
| Les techniques de programmation qui comprennent : la programmation linéaire, la programmation structurée (avantages et constructions de la programmation structurée), la conception modulaire des programmes. | Técnicas de programação que incluem: programação linear, programação estruturada (vantagens e construções da programação estruturada), design de programa modular. |
| Les lectures peuvent être obtenues dans la liste des livres fournis dans la section référence pour cette unité et des informations connexes et pertinentes sur Internet. | As leituras podem ser obtidas na lista de livros fornecida na seção de referência para esta unidade e informações relacionadas e relevantes na Internet. |
| Par la suite, écrire de brèves notes pour : | Em seguida, escreva notas curtas para: |
| Décrire les techniques de conception de descendante et ascendante. | Descrever técnicas de design de cima para baixo e de baixo para cima. |
| Décrire le processus de développement d'un logiciel. | Descreva o processo de desenvolvimento de software. |
| Décrire les techniques de programmation. | Descreva técnicas de programação. |
| Activité pratique | Atividade prática |
| En utilisant un crayon and du papier ou un ordinateur, construire un organigramme pour l'un des deux problèmes listé dans l'activité 1. | Usando papel e caneta ou um computador, construa um fluxograma para um dos dois problemas listados na Atividade 1. |
| Ecrire un algorithme pour l'organigramme que vous avez construit ci-dessus | |
| Testez votre algorithme utilisant la technique de vérification desk checking | Teste seu algoritmo usando a técnica de verificação de mesa |
| Conclusion | Conclusão |
| Cette activité équipe l'apprenant des techniques de conception de solution. | Essa atividade prepara o aluno com técnicas de design de soluções. |
| L'apprenant acquiert également une expérience pratique sur la façon de concevoir des solutions en utilisant des organigrammes et des algorithmes/pseudocodes | O aluno também ganha experiência prática sobre como projetar soluções usando fluxogramas e algoritmos/pseudocódigos |
| Évaluation | |
| Qu'entendez-vous par le mot conception de programme? | O que você quer dizer com a palavra design de programa? |
| Nommer et décrire deux techniques de résolution de problèmes. | Identifique e descreva duas técnicas de resolução de problemas. |
| Décrire la différence entre les deux techniques de résolution de problèmes mentionnées ci-dessus (2) | Descreva a diferença entre as duas técnicas de resolução de problemas mencionadas acima (2) |

| | |
|--|--|
| Les algorithmes peuvent être présentés par l'utilisation de pseudo-code ou un organigramme. | Algoritmos podem ser apresentados pelo uso de pseudo-código ou fluxograma. |
| Quelle est la différence entre un algorithme et un pseudo-code. | Qual é a diferença entre um algoritmo e um pseudo-código. |
| Le pseudocode est indépendant d'un langage. | O pseudocódigo é independente de um idioma. |
| Qu'est-ce que cela veut dire? | O que isso significa? |
| Quelle est la signification de la documentation dans le processus de résolution de problème? | Qual é o significado da documentação no processo de resolução de problemas? |
| Les algorithmes qui sont développés en utilisant trois constructions de base sont plus faciles à suivre. | Algoritmos desenvolvidos usando três construções básicas são mais fáceis de seguir. |
| Décrire ces constructions. | Descreva essas construções. |
| Décrire les caractéristiques d'un bon algorithme | Descreva as características de um bom algoritmo |
| Quels sont les avantages des organigrammes sur les algorithmes | Quais são as vantagens dos fluxogramas em relação aos algoritmos |
| Nommez les règles pour construire un organigramme | Nomeie as regras para a criação de um organograma |
| La conception des programmes définit la structure du code du programme. | O design do programa define a estrutura do código do programa. |
| Décrire trois techniques de programmation | Descrever três técnicas de programação |
| Quels sont certains des avantages de la programmation structurée | Quais são alguns dos benefícios da programação estruturada |
| Décrivez trois éléments de la programmation structurée | Descrever três elementos da programação estruturada |
| Un algorithme peut être testé en utilisant la technique de vérification appelée desk checking et la technique appelée walkthrough. | Um algoritmo pode ser testado usando a técnica de verificação chamada verificação de mesa e a técnica chamada passo a passo. |
| Décrire ces approches de tests. | Descreva essas abordagens de teste. |
| Principes de Programmation | |
| Unité 2. | |
| Résolution de problème basée sur l'informatique | |
| Activité 2.3 - Activité de recherche en groupe | Atividade 2.3 - Atividade de Pesquisa em Grupo |
| Introduction | Introdução |
| Cette activité consiste à faire en groupes ou en collaboration de la recherche sur certains aspects de la programmation. | Esta atividade consiste na realização de pesquisas sobre certos aspectos da programação em grupos ou em colaboração. |
| L'activité nécessite que l'apprenant obtienne à partir de programmeurs pratiquants le genre de problèmes qu'ils résolvent, les stratégies qu'ils appliquent et les étapes suivies pour parvenir à une solution et les opérations communes à tous les programmes. | A atividade exige que o aluno obtenha dos programadores praticantes os tipos de problemas que resolvem, as estratégias que aplicam e as etapas tomadas para chegar a uma solução e as operações comuns a todos os programas. |
| Ils sont tenus de connaître le réalisme des différentes techniques de résolution de problèmes et leur signification. | Eles são obrigados a conhecer o realismo das várias técnicas de resolução de problemas e seu significado. |
| Détails de l'activité | Detalhes da atividade |
| Dans cette activité, vous devez : | Nesta atividade, você precisa: |
| Vous renseigner auprès d'un développeur sur le genre de problèmes qu'il résout de façon quotidienne. | Aprenda com um desenvolvedor sobre os tipos de problemas que eles resolvem diariamente. |

| | |
|---|--|
| Utilisez les exemples réunis pour vous aider à réfléchir sur un programme beaucoup plus vaste (et pas trop grand) et en appliquant les concepts appris dans cette unité. | Use os exemplos coletados para ajudá-lo a refletir sobre um programa muito maior (e não muito grande) e aplicar os conceitos aprendidos nesta unidade. |
| Préciser ou formuler le problème et le définir en montrant les personnes impliquées, les entrées, les opérations et la sortie. | Especifique ou formule o problema e defina-o mostrando as pessoas envolvidas, as entradas, as operações e a saída. |
| Concevoir la solution au problème en utilisant | Projete a solução para o problema usando |
| . la technique descendante et | . técnica de cima para baixo e |
| . la technique ascendante. | . a técnica de baixo para cima. |
| Utiliser les organigrammes et les pseudocodes / algorithmes pour représenter la solution. | Use fluxogramas e pseudocódigos/algoritmos para representar a solução. |
| Utiliser la technique de vérification desk checking pour tester votre pseudocode / algorithme. | Use a técnica de verificação de mesa para testar seu pseudocódigo/algoritmo. |
| Collaborer avec un autre groupe pour échanger vos solutions conçues (Algorithmes) puis effectuer une vérification par les pairs (walkthrough). | Colabore com outro grupo para trocar suas soluções projetadas (algoritmos) e, em seguida, realizar uma revisão por pares (passo a passo). |
| Conclusion | Conclusão |
| Grâce à cette activité, l'apprenant sera en mesure d'obtenir plus de connaissances sur la résolution de problèmes avec l'ordinateur. | Por meio dessa atividade, o aluno poderá obter mais conhecimento sobre a solução de problemas com o computador. |
| En collaborant avec d'autres, l'apprenant sera capable d'améliorer ses compétences sur les applications des concepts de résolution de problèmes. | Ao colaborar com outras pessoas, o aluno será capaz de melhorar suas habilidades na aplicação de conceitos de resolução de problemas. |
| Évaluation | |
| Nommer les opérations qui sont communes à toutes les solutions | Identificar as operações que são comuns a todas as soluções |
| Outre l'entrée, les opérations et la sortie, quelles autres caractéristiques | Além de entrada, operações e saída, quais outros recursos |
| considérez-vous de la définition d'un problème? | you consider when defining a problem? |
| Décrire une propriété d'un problème bien posé | Describe a property of a well-posed problem |
| Quels sont les avantages de la technique descendante sur la technique ascendante | What are the advantages of the top-down technique over the bottom-up technique |
| Décrivez les résultats que vous avez obtenus à partir de la technique de vérification desk checking. | Describe the results you obtained from the desk checking verification technique. |
| Quelle est la signification de tester un algorithme? | What is the meaning of testing an algorithm? |
| Quels bénéfices avez-vous eu en collaborant avec vos coéquipiers et les membres d'une autre équipe? | What benefits did you have when collaborating with your teammates and members of another team? |
| Résumé de l'unité | Summary of the unit |
| Cette unité a discuté des stratégies de résolution de problème avec l'ordinateur, les étapes suivies lors de l'élaboration d'un programme, les techniques de conception qui comprennent l'approche descendante et l'approche ascendante, les algorithmes et les deux façons de présenter un algorithme ; le pseudocode et l'organigramme. | This unit discussed problem-solving strategies with the computer, the steps followed in developing a program, design techniques that include top-down and bottom-up approaches, algorithms and the two ways to present an algorithm; pseudocode and flowchart. |

| | |
|---|---|
| Il a également été examiné les moyens de tester un algorithme ; vérification desk checking et technique walkthrough (contrôle par les pairs), et un aperçu des techniques de programmation. | Ele também analisou maneiras de testar um algoritmo; verificação de mesa e técnica passo a passo, e uma visão geral das técnicas de programação. |
| Évaluation de l'unité | Avaliação da unidade |
| Vérifiez votre compréhension! | Verifique sua compreensão! |
| Questions | |
| Un organigramme est (2 points) | Um organograma é (2 pontos) |
| Nous avons besoin d'ordinateurs pour résoudre les problèmes en raison de : | Precisamos de computadores para solucionar problemas devido a: |
| (2 points) | (2 pontos) |
| la vitesse dans l'exécution des tâches de calcul | velocidade na execução de tarefas de cálculo |
| moins sujettes aux erreurs | menos propenso a erros |
| réduction de la paperasse | reduzindo a burocracia |
| polyvalence des opérations effectuées | versatilidade das operações realizadas |
| tout ce qui précède | todas as opções acima |
| a, b et c | a, b e c |
| Considérons le système utilisé pour contrôler l'arme nucléaire. | Considere o sistema usado para controlar armas nucleares. |
| L'exposition des êtres humains à cet environnement dangereux n'est pas conseillée. | A exposição de humanos a esse ambiente perigoso não é recomendada. |
| De plus la précision de l'opération en cause est obligatoire comme le système est très critique par nature. | Além disso, a precisão da operação em questão é obrigatória, pois o sistema é muito crítico por natureza. |
| Quels sont les avantages d'un système informatique dans un tel scénario. | Quais são as vantagens de um sistema de computador nesse cenário. |
| (2 points) | |
| précision de l'opération | precisão da operação |
| sécurité du genre humain | segurança da raça humana |
| disponibilité du système | disponibilidade do sistema |
| aucune de ces réponses | nenhuma dessas respostas |
| tout ce qui précède | |
| La définition du problème implique (2 points) | Definir o problema envolve (2 pontos) |
| Identification des personnes clés impliquées | Identificar as pessoas-chave envolvidas |
| Identification des entrées | Identificando entradas |
| Spécification de la sortie de la solution | Especificando a saída da solução |
| Spécification des opérations du programme | Especificando as operações do programa |
| b, c et d | b, c e d |
| tout ce qui précède | |
| Simplicité, clarté et élégance sont les maîtres mots des bons programmes, mais diverses questions de mise en œuvre doivent être prises en considération lors de la conception d'une solution. | Simplicidade, clareza e elegância são as características de bons programas, mas uma variedade de questões de implementação precisam ser consideradas ao projetar uma solução. |
| Lesquels ? | Quais? |
| (2 points) | |
| minimiser les besoins en mémoire | minimizar os requisitos de memória |
| maximiser la lisibilité de la sortie | maximizar a legibilidade da saída |
| maximiser la lisibilité du texte d'origine code source | maximizar a legibilidade do código-fonte do texto original |

| | |
|--|--|
| réduire le temps de développement | reduzir o tempo de desenvolvimento |
| a et c | a e c |
| tous | tutti |
| Faites correspondre les notations d'organigramme suivantes avec leurs fonctions et noms corrects. (6 points) | Combine as seguintes notações de organograma com seus papéis e nomes corretos. (6 pontos) |
| reliant deux étapes | conectando duas etapas |
| traitement | tratamento |
| représentent une condition | representam uma condição |
| bouclage | rodopiando |
| prend des valeurs d'entrée/sortie connecteur | leva o conector de valores de entrada/saída |
| Effectuer calculs/traitement | Realizar cálculos/processamento |
| entrée / sortie | entrada/saída |
| Boucle pour une série d'étapes jusqu'à ce que décision | Faça um loop para uma série de etapas até governante |
| la condition soit satisfaite | a condição seja satisfeita |
| Indiquer le début/fin d'un processus | Indique o início/fim de um processo |
| début / fin | início/fim |
| Instructions | Instruções |
| Le but de cette évaluation est de vérifier les progrès de l'apprenant en déterminant ce que l'apprenant a appris dans cette unité. | |
| Les questions couvrent tout ce qui a été présenté dans cette unité afin d'évaluer la compréhension globale. | |
| Répondez attentivement et si votre score tombe en dessous de 40%, refaire les lectures. | |
| entre 40% et 60%, refaire les lectures sur votre zone de faiblesse | |
| supérieur à 60%, vous avez une quantité importante de connaissances | |
| Système de notation | Sistema de classificação |
| Devoirs 20% | Dever de casa 20% |
| Test d'évaluation 10% | Teste de avaliação 10% |
| Examen final 70% | Exame final 70% |
| Réponses | Respostas |
| Un organigramme est une représentation schématique d'un algorithme | Um fluxograma é uma representação esquemática de um algoritmo |
| a. indiquer le début et la fin d'un processus <i>début / fin</i> | a. indicar o início e o fim de um processo de <i>início/término</i> |
| effectuer des calculs et le traitement | realizar cálculos e processamento |
| <i>traitement</i> | |
| représentent une condition | |
| prise de décision | tomada de decisão |
| boucle pour une série d'étapes jusqu'à ce que la condition | loop para uma série de etapas até a condição |
| soit satisfaite | fique satisfeito |
| <i>bouclage</i> | |
| prenant des valeurs d'entrée/sortie | obtendo valores de entrada/saída |
| <i>entrée / sortie</i> | |

| | |
|--|---|
| reliant deux étapes | |
| <i>connecteur</i> | <i>conectora</i> |
| Unité 3. | Unidade 3. |
| Programmation en langage C | Programação na linguagem C |
| Introduction de l'unité | Introdução da unidade |
| Les concepts de la programmation peuvent être différents selon les langages, mais quelques instructions de base apparaissent dans presque tous les langages. | Os conceitos de programação podem ser diferentes em linguagens diferentes, mas algumas instruções básicas aparecem em quase todas as linguagens. |
| Ces instructions incluent les instructions d'entrée qui sont utilisés pour obtenir des données à partir du clavier, un fichier, ou tout autre dispositif; les instructions de sortie utilisées pour afficher des informations à l'écran ou envoyer des informations dans un fichier ou tout autre dispositif; les instructions arithmétiques effectuant des opérations arithmétiques de base comme l'addition et la multiplication; les instructions conditionnelles utilisées pour vérifier certaines conditions et exécuter la séquence appropriée des déclarations et les instructions de répétition effectuant une action à plusieurs reprises, le plus souvent avec une certaine variation. | Essas instruções incluem instruções de entrada que são usadas para obter dados do teclado, um arquivo ou qualquer outro dispositivo; instruções de saída usadas para exibir informações na tela ou enviar informações para um arquivo ou qualquer outro dispositivo; instruções aritméticas executando operações aritméticas básicas, como adição e multiplicação; instruções conditionais usadas para verificar certas condições e executar a sequência adequada de declarações; e repetir instruções executando uma ação repetidamente, geralmente com alguma variação. |
| Par conséquent, cette unité traite les concepts de programmation comme les variables, les types de données, les opérateurs, les structures de contrôle, les tableaux et les procédures ou fonctions. | Portanto, esta unidade lida com conceitos de programação, como variáveis, tipos de dados, operadores, estruturas de controle, matrizes e procedimentos ou funções. |
| Les procédures ou fonctions renforceront le concept de décomposition d'un problème en problèmes plus petits problèmes (techniques de conception de haut en bas et / ou de bas en haut). | Procedimentos ou funções reforçarão o conceito de dividir um problema em problemas menores (técnicas de design de cima para baixo e/ou de baixo para cima). |
| Des programmes simples seront écrits en langage C pour renforcer toutes les notions apprises dans l'unité précédente. | Programas simples serão escritos em linguagem C para reforçar todos os conceitos aprendidos na unidade anterior. |
| Objectifs de l'unité | Objetivos da unidade |
| À la fin de cette unité, vous devriez être capable de : | |
| Télécharger et installer un programme. | Baixe e instale um programa. |
| Distinguer les différents types de données. | Distinguir os diferentes tipos de dados. |
| Fournir des solutions en écrivant des programmes simples en C. | Forneça soluções escrevendo programas simples em C. |
| Unité 3. | Unidade 3. |
| Programmation en langage C | Programação na linguagem C |
| Unité 3. | |
| Programmation en langage C | |
| Principes de Programmation | |
| Principes de Programmation | |
| Termes clés | |

| | |
|--|---|
| Type de données : types de données définissent la manière dont les valeurs et les choix de valeurs sont représentées dans un système | Tipo de dados: os tipos de dados definem como os valores e as opções de valor são representados em um sistema |
| Variables : ce sont des noms donnés à la zone de mémoire de stockage. | Variáveis: são nomes dados à área de memória de armazenamento. |
| La valeur d'une variable est modifiée pendant l'exécution du programme. | O valor de uma variável é alterado durante a execução do programa. |
| Une constante : | Uma constante: |
| Représente une valeur stockée dans la mémoire et qui ne peut pas changer. | Representa um valor que é armazenado na memória e não pode ser alterado. |
| Déclaration de variable : déclare le type de la variable et alloue de la mémoire pour cette variable. | Declaração de variável: declara o tipo da variável e aloca memória para essa variável. |
| Un identifiant : | Um identificador: |
| C'est un nom de variable | É um nome de variável |
| Application Console : | Aplicativo de console: |
| Une application de la console est un programme informatique conçu pour être utilisé par une interface de l'ordinateur en mode texte, comme un terminal texte, une interface en ligne de commande de certains systèmes d'exploitation (Unix, DOS, etc). | Um aplicativo de console é um programa de computador projetado para ser usado por uma interface de computador baseada em texto, como um terminal de texto, uma interface de linha de comando de certos sistemas operacionais (Unix, DOS, etc.). |
| Environnement de développement intégré (IDE) : | Ambiente de Desenvolvimento Integrado (IDE): |
| Un environnement de développement intégré (IDE) est une application logicielle qui fournit des installations complètes pour les programmeurs informatiques de développement de logiciels. | Um Ambiente de Desenvolvimento Integrado (IDE) é um aplicativo de software que fornece instalações completas para programadores de computador de desenvolvimento de software. |
| Code source : | Código fonte: |
| C'est un programme de l'ordinateur écrit dans un langage de haut niveau qui est converti en code objet ou code machine par un compilateur. | É um programa de computador escrito em uma linguagem de alto nível que é convertido em código objeto ou código de máquina por um compilador. |
| Types de données dérivées : | Tipos de dados derivados: |
| Ils sont définis en fonction d'autres types de données, appelés types de base. | Eles são definidos com base em outros tipos de dados, chamados de tipos base. |
| Les types dérivés peuvent avoir des attributs, et peuvent avoir des éléments ou un contenu mixte (http://msdn.microsoft.com). | Os tipos derivados podem ter atributos e podem ter elementos ou conteúdo misto (http://msdn.microsoft.com). |
| Types de données primitifs : les types de données primitifs ne sont pas définis en fonction d'autres types de données, car ils constituent la base de tous les autres types. | Tipos de dados primitivos: os tipos de dados primitivos não são definidos com base em outros tipos de dados porque formam a base de todos os outros tipos de dados. |
| Ils ne peuvent pas avoir des contenus d'éléments ou des attributs (http://msdn.microsoft.com). | Eles não podem ter conteúdo ou atributos de elementos (http://msdn.microsoft.com). |
| Constante Cadeia : c'est une chaîne de caractères entre guillemets. | Constante de cadeia: é uma cadeia de caracteres entre aspas. |
| (Fondamentale) | (Fundamental) |
| Unité 3. | |
| Programmation en langage C | |
| Unité 3. | |

| | |
|--|---|
| Programmation en langage C | |
| Un opérateur : | Um operador: |
| C'est un symbole qui permet à l'utilisateur de demander à l'ordinateur de faire un certain nombre de calcul mathématique ou logique. | É um símbolo que permite ao usuário pedir ao computador para fazer um certo número de cálculos matemáticos ou lógicos. |
| (Fondamentale) | (Fundamental) |
| Types de données définis par l'utilisateur : | Tipos de dados definidos pelo usuário: |
| Ils sont définis par l'utilisateur en fonction des types de données existants. | Eles são definidos pelo usuário com base nos tipos de dados existentes. |
| Éditeur de code source : | Editor de código-fonte: |
| Un éditeur de code source est un éditeur de programme basé sur le texte et spécialement conçu pour l'édition de code source des programmes informatiques par les programmeurs. | Um editor de código-fonte é um editor de programa baseado em texto especialmente projetado para a edição de código-fonte de programas de computador por programadores. |
| Il peut être une application autonome ou il peut être intégré dans un environnement intégré de développement (IDE) ou navigateur Web. | Ele pode ser um aplicativo independente ou pode ser integrado a um ambiente de desenvolvimento integrado (IDE) ou navegador da Web. |
| (http://en.wikipedia.org/wiki/Source code editor) | (editor de código http://en.wikipedia.org/wiki/Source) |
| Fichiers d'en-tête : | Arquivos de cabeçalho: |
| Les fichiers d'en-tête contiennent des définitions de fonctions et de variables qui peuvent être incorporés dans n'importe quel programme C en utilisant l'instruction de préprocesseur, #include (http:// gd.tuwien.acat/languages/c/programming-bbrown/_ c 011.htm). | Os arquivos de cabeçalho contêm definições de funções e variáveis que podem ser incorporadas em qualquer programa C usando a instrução de pré-processor, #include (http://gd.tuwien.acat/languages/c/programming-bbrown/_ c 011.htm). |
| Bibliothèque : | Biblioteca: |
| Une bibliothèque est une collection de programmes (généralement) précompilés, réutilisables qu'un programmeur peut « appeler » lors de l'écriture de son code afin qu'il n'ait pas besoin de les réécrire (searchsqlserver.techtarget.com/definition/library). | Uma biblioteca é uma coleção de programas (geralmente) pré-compilados e reutilizáveis que um programador pode "chamar" ao escrever seu código para que não precise reescrevê-los (searchsqlserver.techtarget.com/definição/library). |
| Une bibliothèque en C est un groupe de fonctions et de déclarations, exposées pour une utilisation par d'autres programmes. (http://en.wikibooks.org/ wiki/C Programming/Libraries). | Uma biblioteca C é um grupo de funções e declarações, expostas para uso por outros programas. (http://en.wikibooks.org/ Programação/Bibliotecas Wiki/C). |
| Un commentaire : | Um comentário: |
| Un "commentaire" est une séquence de caractères commençant par une combinaison de slash/astérisque (/*) et se terminant par un astérisque/ slash (*/) qui est considérée comme un seul caractère d'espace blanc par le compilateur et est ignorée. | Um "comentário" é uma sequência de caracteres começando com uma combinação de barra/asterisco (/*) e terminando com um asterisco/barra (*/) que é tratado como um único caractere de espaço em branco pelo compilador e é ignorado. |
| Il existe également des commentaires en ligne précédés par deux barres obliques (//) (http:// msdn.microsoft.com/ en-us/library/wfwd74e.aspx). | Também há comentários embutidos precedidos por duas barras (//) (http://msdn.microsoft.com/ en-us/library/wfwd74e.aspx). |
| Principes de Programmation | |

| | |
|--|---|
| Principes de Programmation | |
| Une expression : | Uma expressão: |
| Une expression est une combinaison de variables, des constantes, et d'opérateurs conformément à la syntaxe du langage (livre par ISRD) | Uma expressão é uma combinação de variáveis, constantes e operadores de acordo com a sintaxe da linguagem (livro de ISRD) |
| Type de casting : | Tipo de elenco: |
| La notion transtypage en langage C est utilisée pour modifier le type d'une variable. | A noção de tipagem na linguagem C é usada para modificar o tipo de uma variável. |
| Le nouveau type de données doit être mentionné avant le nom de variable entre parenthèses (http://fresh2refresh.com/c/c-type-casting). | O novo tipo de dados deve ser mencionado antes do nome da variável entre parênteses (http://fresh2refresh.com/c/c-type-casting). |
| Portée des règles : | Âmbito das regras: |
| C'est une région du programme où une variable définie peut avoir son existence et au-delà de laquelle cette variable ne peut être accessible. | É uma região do programa onde uma variável definida pode existir e além da qual essa variável não pode ser acessada. |
| On distingue les variables locales, les variables globales et les paramètres formels (http://www.tutorialspoint.com/cprogramming/c_scope_rules.html). | Distinguimos entre variáveis locais, variáveis globais e parâmetros formais (http://www.tutorialspoint.com/cprogramming/c_scope_rules.html). |
| Paramètres formels : | Parâmetros formais: |
| Ce sont des paramètres qui agissent comme des espaces réservés pour les paramètres réels. | Esses são parâmetros que atuam como espaços reservados para os parâmetros reais. |
| Paramètres effectifs : | Configurações efetivas: |
| Ce sont les paramètres (ou valeurs) transmis aux paramètres réels de la fonction d'appel (environnement). | Estes são os parâmetros (ou valores) passados para os parâmetros reais da função de chamada (ambiente). |
| Les variables locales : | Variáveis locais: |
| Les variables qui sont déclarées dans une fonction ou d'un bloc sont appelées variables locales. | As variáveis declaradas em uma função ou bloco são chamadas de variáveis locais. |
| Ils peuvent être utilisés que par des déclarations qui sont à l'intérieur de cette fonction ou de bloc de code. (http://www.tutorialspoint.com/cprogramming/c_scope_rules.htm) | Eles só podem ser usados por instruções que estão dentro dessa função ou bloco de código. (http://www.tutorialspoint.com/cprogramming/c_escopo_rules.htm) |
| Les variables globales : | Variáveis globais: |
| Les variables globales sont définies en dehors d'une fonction, généralement en haut dans le programme. | As variáveis globais são definidas fora de uma função, geralmente na parte superior do programa. |
| Les variables globales gardent leur valeur pendant toute la durée d'exécution de votre programme et peuvent être consultée à l'intérieur de l'une des fonctions. (http://www.tutorialspoint.com/cprogramming/c_scope_rules.htm) | As variáveis globais mantêm seu valor durante a execução do seu programa e podem ser visualizadas em uma das funções. (http://www.tutorialspoint.com/cprogramming/c_escopo_rules.htm) |
| Activités d'apprentissage | Atividades de aprendizagem |
| Activité 3.1 - Types de données, variables et opérateurs | Atividade 3.1 - Tipos de dados, variáveis e operadores |
| Introduction | Introdução |
| Afin de discuter et d'illustrer ces concepts à l'aide d'un programme C, nous devons d'abord commencer par nous familiariser avec | Para discutir e ilustrar esses conceitos usando um programa em C, devemos primeiro nos familiarizar |

| | |
|---|---|
| l'environnement de développement (aperçu du langage C). | com o ambiente de desenvolvimento (visão geral da linguagem C). |
| Il y a de bonnes références qui constituent un bon point de départ pour la programmation C, y compris les références fournies pour cette unité. | Existem algumas boas referências que são um bom ponto de partida para a programação C, incluindo as referências fornecidas para esta unidade. |
| Programming and problem solving through 'C' languages by Harsha Priya, r. Ranjeet, Fundamentals of programming languages by Dipali P. Baviskar and Programming and PROB solving using C. by ISRD. | Programação e resolução de problemas através de linguagens 'C' por Harsha Priya, r. Ranjeet, Fundamentos de linguagens de programação por Dipali P. Baviskar e Programação e resolução de PROB usando C. por ISRD. |
| Beaucoup de compilateurs C existent sur le marché et ont été conçus ou développés par différents groupes de développeurs de programmes. | Muitos compiladores C existem no mercado e foram projetados ou desenvolvidos por diferentes grupos de desenvolvedores de programas. |
| Par conséquent, la préférence d'un compilateur est laissée à vous (apprenant). | Portanto, a preferência de um compilador é deixada para você (aluno). |
| La préférence de ce cours est le compilateur Dev C++. | A preferência para este curso é o compilador C++ Dev. |
| Vous pouvez le télécharger, à partir de http://liquidtelecom.dl.sourceforge.net (Dev-cpp 5.8.3 TDM-gcc 4.8.1 setup.exe) ou la dernière version mise à jour qui peut fonctionner sur la dernière version de Windows. | Você pode baixá-lo em http://liquidtelecom.dl.sourceforge.net (Dev-Cpp 5.8.3 tdm-gcc 4.8.1 setup.exe) ou a versão atualizada mais recente que pode ser executada na versão mais recente do Windows. |
| Si vous n'êtes pas en mesure de le récupérer, demandez à votre instructeur de vous aider. | Se você não conseguir recuperá-lo, peça ajuda ao seu instrutor. |
| Le compilateur lit c ++ et cela ne devrait pas être un souci pour vous car vous allez apprendre comment compiler un programme C en renommant les fichiers pour avoir l'extension C (.c). | O compilador lê c ++ e isso não deve ser uma preocupação para você, pois você aprenderá a compilar um programa C renomeando os arquivos para ter a extensão C (.c). |
| Avant de créer le fichier du code source, il est nécessaire de créer un projet (Fichier> Nouveau> Projet) et les options comprennent l'application de la console, le nom du projet et projet C pour le langage C. | Antes de criar o arquivo de código-fonte, é necessário criar um projeto (Arquivo> Novo> Projeto) e as opções incluem aplicativo de console, nome do projeto e projeto C para linguagem C. |
| L'activité suivante consiste à sélectionner le dossier pour stocker les fichiers, dans la fenêtre suivante. | A próxima atividade é selecionar a pasta para armazenar os arquivos na próxima janela. |
| Après avoir indiqué le dossier où le fichier de configuration du projet (.dev) sera sauvegardé, l'IDE génère un fichier source de base (par défaut, main.c). | Depois de especificar a pasta onde o arquivo de configuração do projeto (.dev) será salvo, o IDE gera um arquivo de código-fonte base (por padrão, main.c). |
| La fenêtre IDE comprend trois sous-fenêtres : l'explorateur de fichier du projet, les onglets de résultat, et l'éditeur de code source. | A janela do IDE consiste em três subjanelas: o explorador de arquivos do projeto, as guias de resultados e o editor de código-fonte. |
| Ces fenêtres peuvent être redimensionnées et minimisés. | Essas janelas podem ser redimensionadas e minimizadas. |
| | |
| L'instruction system("PAUSE") ; est automatiquement inclus pour mettre en pause l'exécution du programme avant de fermer la fenêtre pour permettre de voir les résultats du programme. | A instrução do sistema ("PAUSE"); é automaticamente incluída para pausar a execução do programa antes de fechar a janela para permitir a visualização dos resultados do programa. |

| | |
|---|---|
| La fenêtre Explorateur de Fichiers (Files Explorer) affiche le nom du projet et les fichiers inclus. | A janela Explorador de Arquivos exibe o nome do projeto e os arquivos incluídos. |
| L'onglet Projet (Project tab) contient généralement un seul fichier avec le code source du programme. | A guia Projeto geralmente contém um único arquivo com o código-fonte do programa. |
| Dans ce volet, nous pouvons trouver deux onglets supplémentaires : | Neste painel, podemos encontrar duas guias adicionais: |
| Classes et Debug. | Classes e depuração. |
| L'onglet Classes affiche les fonctions du programme tandis que l'onglet Debug montre des variables visualisées durant le processus de débogage. | A guia Classes exibe as funções do programa, enquanto a guia Depurar mostra as variáveis que são visualizadas durante o processo de depuração. |
| La fenêtre des résultats (Results Windows) est utilisée pour présenter les résultats des actions de l'IDE : les erreurs de compilation, les directives de compilation, les commandes de débogage. | A janela Resultados é usada para apresentar os resultados das ações do IDE: erros de compilação, diretivas de compilação, comandos de depuração. |
| L'éditeur de code source affiche le code du programme. | O editor de código-fonte exibe o código do programa. |
| Une fois le projet a été créé, nous pouvons commencer à écrire puis exécuter notre programme C. Par exemple, le programme simple, Hello world! ci-dessous. | Depois que o projeto for criado, podemos começar a escrever e depois executar nosso programa C. Por exemplo, o programa simples, Hello World! abaixo. |
| Principes de Programmation | |
| Principes de Programmation | |
| Unité 3. | |
| Programmation en langage C | |
| Unité 3. | |
| Programmation en langage C | |
| Unité 3. | |
| Programmation en langage C | |
| Le procédé d'exécution d'un programme en C comporte les étapes suivantes : | O método para executar um programa em C compreende as seguintes etapas: |
| Création du programme | Criando o programa |
| Compilation du programme | Compilando o programa |
| Lier le programme avec les fonctions qui sont nécessaires à partir de la | Vinculando o programa com as funções que são exigidas do |
| bibliothèque C | Biblioteca C |
| L'exécution du programme | A execução do programa |
| Un résumé de ces étapes peut être représenté par l'organigramme ci-après : | Um resumo dessas etapas pode ser representado pelo seguinte fluxograma: |
| Code Programme | Código do programa |
| Système prêt | Pronto para sistema |
| Entrer Programme | Entrar no programa |
| Programme Source | Programa de origem |
| Compiler Programme Source | Programa de compilação de origem |
| Éditer Programme Source | Editar programa de origem |
| Compilateur C | Compilador C |
| Erreurs de syntaxe ? | Erros de sintaxe? |
| Oui | sim |
| Librairies système | Bibliotecas do sistema |
| Liens avec les librairies systèmes | Links com bibliotecas do sistema |
| Exécuter code Objet | Executar código de objeto |

| | |
|---|--|
| run this program using the console pauser or add your own getch, system("pause") or input loop <*/< | execute este programa usando o pauser do console ou adicione seu próprio getch, sistema ("pausa") ou loop de entrada*/ |
| un commentaire< | um comentário< |
| Fonction principale | Função principal |
| int main(int argc, char "argv") | int main (int argc, char "argv") |
| { printf("hello"); | {printf ("olá"); |
| return 0; | retorno 0; |
| Corps du programme | Corpo do programa |
| contenant les instructions | contendo instruções |
| exécutables contenues entre '{ }'. | executáveis contidos entre '{}'. |
| } | } |
| Structure d'un programme C | Estrutura de um programa C |
| Voir figure ci-dessous pour un programme en utilisant la version 5.6.2 de Dev C ++ | Veja a figura abaixo para um programa usando Dev C++ versão 5.6.2 |
| Fonctions printf () et scanf () | funções printf () e scanf () |
| La fonction printf () est utilisée pour écrire des informations dans un fichier, à l'écran ou tout autre périphérique de sortie. | A função printf () é usada para gravar informações em um arquivo, na tela ou em qualquer outro dispositivo de saída. |
| Tandis que la fonction scanf () est utilisée pour lire les données que le programme manipule ou écrit à l'écran. | Enquanto a função scanf () é usada para ler dados que o programa manipula ou grava na tela. |
| Ces fonctions sont prises en charge par le fichier d'en-tête ou directive du préprocesseur <stdio.h>, ce qui signifie fichiers d'entête d'entrée et de sortie standard. | Essas funções são suportadas pelo cabeçalho do pré-processor ou arquivo de diretiva<stdio.h>, o que significa arquivos de cabeçalho de entrada e saída padrão. |
| Par exemple, un programme qui affiche une valeur qui est déjà affectée à une variable et une valeur (age) donnée le système par l'utilisateur. | Por exemplo, um programa que exibe um valor que já está atribuído a uma variável e um valor (idade) dado ao sistema pelo usuário. |
| Le programme invite l'utilisateur à entrer son âge : | O programa solicita que o usuário insira sua idade: |
| #include <stdio.h> | #include <stdio.h> |
| #include <stdlib.h> | #include <stdlib.h> |
| /* Exécuter ce programme en utilisant le system("pause") ou une boucle d'entrée */ | /* Execute este programa usando o sistema ("pausa") ou um loop de entrada */ |
| int main (int argc, char * argv []) { | int principal (int argc, char * argv []) { |
| int y = 30; // Valeur directement affectée à la variable | int y = 30; //Valor atribuído diretamente à variável |
| int age; | Int idade |
| printf ("Entrez votre âge \ n"); // Demander à l'utilisateur de lire son | printf ("Digite sua idade\ n"); //Peça ao usuário para ler seus |
| âge | idade |
| scanf("%d",&age); // Lecture de l'âge en tapant les valeurs sur le | scanf ("%d", &age); //Ler a idade digitando valores na |
| clavier | teclado |