**Língua inglesa – Software Development**

|  |
| --- |
| **Objectives:**  To encourage students to use English-English Internet dictionaries. To reinforce the habit of using a monolingual dictionary. To help learners be more autonomous in their learning.  To encourage students to search the internet as part of their work. To read intensively to get a more detailed understanding of a topic. To clearly communicate ideas and produce well-structured arguments. |

EXERCÍCIOS

**Look at the dictionaries (links below) to help you answer to the questions:**

<https://www.collinsdictionary.com/dictionary/english/>

<https://www.dictionary.com/>

<https://www.merriam-webster.com/>

**Group work**

Read the text and do the activities.

**What is Software Development?**

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and fixing bugs involved in creating and maintaining applications, frameworks, or other software components. In other words, it's the process of making a new software product, from its initial concept to its final release, and beyond.

This software could be a small tool to help with day-to-day tasks, a mobile game, a business process system for large corporations, or anything in between. Software is embedded in countless aspects of our modern lives and plays a key role in numerous industries such as healthcare, finance, entertainment, and more.

**What is the Role of Software Development?**

The role of software development is crucial in today's digital age, where almost everything runs on software. At its core, software development enables people and businesses to solve problems, achieve specific goals, and generally function more effectively. Here are some of the key roles of software development:

**Problem-Solving**: Software can automate repetitive tasks, analyse large amounts of data quickly, improve efficiency, and solve numerous other problems.

**Innovation**: New software can disrupt industries, changing how we work, communicate, and live our lives.

**Enabling Business Processes**: Software is often key to providing goods and services, whether it's a website for an online retailer, an app for a food delivery service, or a complex system managing logistics for a large corporation.

**What is the Software Development Process?**

The software development process is the series of steps that developers follow to create a new software product. While the exact steps can vary, most software development includes the following stages:

**Requirements Analysis**: This initial stage involves understanding what the software needs to do. Developers work with clients or end-users to determine the features and functionality required.

**Design**: Once the requirements are clear, developers create a plan for how the software will work. This can include deciding on the architecture, interfaces, and data models that will be used.

**Coding**: This is the actual writing of the software. Developers create code in a suitable programming language to bring the design to life.

**Testing**: Developers and testers work together to find and fix any issues with the software. They aim to ensure the software works correctly, is easy to use, and doesn't have any bugs.

**Deployment**: Once the software is tested and refined, it's released to the users. This could involve deploying it to a server, publishing it in an app store, or distributing it in another way.

**Maintenance**: After release, developers continue to work on the software. They might add new features, fix any problems that come up, or update the software to keep up with changing technology.

This process isn't always linear - it can be iterative, with developers going back to previous steps as needed. This is especially common in Agile development, where the process is divided into small, manageable parts called "sprints", allowing for continuous improvement and flexibility.

In a nutshell, software development is a creative and technical process that plays a vital role in shaping the way we live and work. It's a process that involves collaboration, problem-solving, and continual learning, and is an exciting field with endless possibilities.

***Activity 1*** – **Developing vocabulary skills**

The function of a word in a sentence is determined by its position and how it relates to other words in the sentence. To enhance your understanding, it's essential to examine and think about the role or 'job' of the underlined words in the text. How are these words contributing to the overall meaning of the sentence? What purpose do they serve?

Choose your favourite online dictionary. Identify the part of speech, meaning, and pronunciation of these vocabulary items. Write an example sentence showcasing the use of these words. Remember, your focus should not just be on the words themselves, but also on what they are doing within those sentences - their function or 'job'.

In the sentence, "Software development is the process of ***conceiving***, specifying, designing, programming, documenting, testing, and fixing bugs…", 'conceiving', 'specifying', 'designing', 'programming', 'documenting', 'testing', and 'fixing' are all gerunds. A gerund is a verb form that ends in "-ing" and works like a noun in the structure of a sentence. We use gerunds when we want to focus on ***the general idea of an activity***, not the specific action. So, when we say 'conceiving' or 'designing', these words are acting like nouns, even though they are based on verbs.

The preposition 'of' usually indicates possession, origin, or composition. So, 'of' in our sentence is used to detail what the process consists of, or is composed of, and its objects are all the steps involved in software development, represented by our gerunds. In simpler terms, if we say that software development is a process, the 'of' tells us more about this process. The process includes all those steps: conceiving an idea, specifying requirements, designing the software, programming it, documenting the work, testing it for errors, and fixing those bugs. Each one of these activities is an integral part of the process 'of' software development.

| **vocabulary item** | | **Part of speech** | **meaning** | **pronunciation** | **example sentence** |
| --- | --- | --- | --- | --- | --- |
| ***1*** | ***Conceiving*** | gerund, verbal noun | Forming or devising a plan or idea in the mind | /kənˈsiːvɪŋ/  /Cãmciivin/ | Gerund: "***Conceiving*** a unique software solution requires a deep understanding of the problem space."  **Gerund as Subject**: "***Conceiving*** *innovative features* can significantly enhance the user experience of the software."  **Gerund as Object**: "The initial phase of software development involves thoroughly ***conceiving*** the concept to ensure its viability and usefulness." |
| ***2*** | maintaining | Verbal noun, gerund | Keeping something in the same state. | / meɪnˈteɪn / | To keep the car working, is necessary to maintaining is condiction.  Maintaining a good habit is the key to success. |
| ***3*** | frameworks | noun | a structure that provides a base for the application development process | /ˈfreɪmwɜːrks/ | Frameworks help writing code faster. |
| ***4*** | Release | Verb, noun | Publishing a new version of something | / rɪˈlis / | The team need to release an updated software.  The release of a new car. |
| ***5*** | Day-to-day | adjective | Something you do every day/ a routine | / ˈdeɪ təˈdeɪ / | Her day-to-day work is delivering goodies. |
| ***6*** | Embedded | Verb,  adjective | Fixed/atatched in something | / ɛmˈbɛd ɪd / | This ship is embedded in the board. |
| ***7*** | Core | Noun,  Adjective,  verb | The centre of something  Something essential | / kɔr / | The core function of this app is showing the notes.  We are in the core of the city. |
| ***8*** | Effectively | adverb |  | / ɪˈfɛk tɪv li, iˈfɛk‐ / |  |
| ***9*** | Automate | verb | Do something with none or almost none human interactions | / ˈɔ təˌmeɪt / | We intend to automate the cooking processes. |
| ***10*** | Efficiency | noun | Do something with minimal wasted time/resources | / ɪˈfɪʃ ən si / | The efficiency of this new power supply is unbeatable. |
| ***11*** | Disrupt | verb | Interrupt something already happening | / dɪsˈrʌpt / | I disrupt the meeting. |
| ***12*** | enabling | Verb,  adjective | Giving the means to do something | / ɛnˈeɪ blɪŋ / | We are enabling him to do this things. |
| ***13*** | providing | Verbal noun, gerund | Giving something | / prəˈvaɪ dɪŋ / | I’m providing the building for the company. |
| ***14*** | End-users | noun | The persons who gone use the final version of something | /ˈɛnd ˈjuːzərz/ | The end-users like the website. |
| ***15*** | Interfaces | noun | Points of interaction between different systems | /ˈɪntərˌfeɪsɪz/ | The interfaces in this app are easy to use. |
| ***16*** | Refined | Verb,  adjective | Make something more polish/pure | / rɪˈfaɪnd / | This program as being refined already.  This is the refined iron ore. |
| ***17*** | deploying | Verbal noun, gerund | Putting something in operation | /dɪˈplɔɪɪŋ/ | The team is deploying a new patch for the system. |
| ***18*** | features | noun | The things a user can use in a program | /ˈfiːtʃərz/ | This app has a lot of new features. |
| ***19*** | Come up | Phrasal verb | Something that occur unexpectedly | /kʌm ʌp/ | Ana come up with a resolution for the problem. |
| ***20*** | iterative | adjective | Making repetitions | / ˈɪt əˌreɪ tɪv, -ər ə tɪv / | Adopt an iterative approach to migration. |
| ***21*** | manageable | adjective | Something we can handle | / ˈmæn ɪ dʒə bəl / | This job is manageable. |
| ***22*** | In a nutshell | Idiomatic expression | Explaining something in a short sentence | /ɪn ə ˈnʌtʃɛl/ | In a nutshell, she won the race. |
| ***23*** | shaping | Verbal noun, gerund | Changing the form of something | ˈʃeɪpɪŋ/ | I’m shaping this piece of wood. |
| ***24*** | Agile development | noun | A methodology in software development that emphasizes flexibility, collaboration, and incremental progress through iterative cycles called sprints. | /ˈædʒaɪl/ | Agile development helps teams adapt quickly to changing requirements. |
| ***25*** | sprints | noun | Doing something fast | /sprɪnts/ | He won all the sprints. |

***Activity 2*** - Choose at least six of the previous vocabulary items and create a 5–8-line text.

***Activity 3*:** **Group search activity**

In this group research activity, each group is assigned a specific topic to research based on the provided text about Software Development. After researching, each group will present their findings to the rest of the class. Be sure to evaluate your sources for credibility and relevance to your topic. Organise your findings logically and prepare a presentation that communicates your findings clearly and effectively.

**Tips for the research**

1. Start with a clear understanding of the assigned topic and its relevance to software development.
2. Use reliable sources to gather information.
3. Focus on finding recent and up-to-date information, as the field of software development is constantly evolving.
4. Use multiple sources to verify facts and gather diverse perspectives on the topic.
5. Take notes, highlighting key points and examples that can help illustrate the topic during the presentation.
6. Organise the gathered information logically, focusing on the most important aspects of the topic.
7. Be prepared to explain technical terms and concepts in simple language to ensure that all audience members can understand the material.
8. Consider using visual aids, to help convey information more effectively.
9. Practice presenting the information, ensuring that it is concise and engaging, and be prepared to answer questions from the audience.

**Specific topic to research**

**Group 1: The Roles and Impact of Software Development in Different Industries**

* Start with an understanding of how software development affects various industries.
* Research case studies of how new software has revolutionised sectors like healthcare, finance, or entertainment.
* Look for statistics that demonstrate the impact of software on these industries.
* Identify the key challenges and opportunities associated with software development in each industry.

**Group 2: Agile Development and Its Influence on Software Development**

* Begin by understanding the basics of Agile development methodology.
* Investigate the principles of Agile development and how it differs from other methodologies like Waterfall.
* Look for real-world examples of software development projects that used Agile methodology and analyse their outcomes.
* Research the benefits and drawbacks of Agile development.

**Group 3: The Software Development Life Cycle (SDLC) in Detail**

* Start by examining each stage of the SDLC: requirements analysis, design, coding, testing, deployment, and maintenance.
* Look for best practices associated with each stage.
* Find case studies or examples of projects, and analyse how they navigated through each of these stages.
* Explore challenges that can occur at each stage and ways to mitigate them.

**Group 4: Problem-Solving in Software Development**

* Begin with understanding the role of problem-solving in software development.
* Research the common problems that software development aims to solve in various contexts (business, personal use, etc.).
* Find examples of software that have been particularly effective in problem-solving.
* Look into how software developers approach problem-solving during the development process.

**Group 5: Emerging Trends and Technologies in Software Development**

* Start with an overview of current emerging trends in software development, such as artificial intelligence, machine learning, blockchain, IoT (Internet of Things), and cloud computing.
* Investigate how these technologies are being integrated into software development practices.
* Look for case studies or examples of software projects leveraging these technologies.
* Discuss the potential future impact of these trends on various industries and the software development process.

**Evaluation Criteria**

You will be assessed both on your group's performance leading up to the presentation and on different aspects of the actual presentation.

**Group's Performance Leading Up to the Presentation**

**Communication:** How well the group shares information and communicates within the team.

**Collaboration:** How effectively the group works together and supports each other’s contributions.

**Task Delegation:** How tasks are assigned within the group to ensure everyone has a role.

**Research:** The thoroughness and depth of the group's research on the topic.

**Information Gathering:** How effectively the group collects necessary data and resources.

**Critical Thinking:** The ability to analyse information and form reasoned conclusions.

**Problem Solving:** How the group addresses and resolves any challenges that arise.

**Time Management:** The group's ability to use time wisely throughout the project preparation.

**Meeting Deadlines:** How well the group adheres to deadlines set for tasks and milestones.

**Presentation Planning:** The process of organising and structuring the presentation content.

**Preparation:** The readiness of the group for the presentation, including rehearsal and material preparation.

**Group Dynamics:** The interaction and relationship dynamics within the group.

**Aspects of the Presentation**

**Introducing the Topic:**

**Clarity:** The introduction should be clear and easy to understand.

**Engagement:** The opening should capture the audience's attention effectively.

**Conciseness:** The opening statement should be brief and to the point.

**Overview of Presentation:**

**Organisation:** The presentation should have a clear structure.

**Explanation:** Clearly explain what will be covered in the presentation.

**Transitions:** Move smoothly from one section to another.

**Beginning a New Section:**

**Introduction:** Each new section should start clearly and engagingly.

**Guidance:** Use signposts to help the audience follow along.

**Finishing a Section:**

**Wrap-up:** Each section should end with a clear summary of key points.

**Transition:** Shift smoothly to the next section or topic.

**Analysing Points and Offering Recommendations:**

**Depth:** Show a deep understanding of the topic in your analysis.

**Recommendations:** Provide thoughtful and actionable suggestions.

**Communication:** Explain complex ideas in a clear and straightforward way.

**Giving Examples:**

**Relevance:** Use examples that clearly relate to and illustrate your points.

**Clarity:** Explain how each example supports the topic.

**Engagement:** Choose interesting and engaging examples.

**Paraphrasing and Clarifying:**

**Rephrasing:** Effectively simplify and clarify complex ideas.

**Conciseness:** Keep explanations brief and to the point.

**Summarising and Drawing Conclusions:**

**Summarising:** Clearly recap the main points at the end of the presentation.

**Conclusions:** Present final thoughts clearly and concisely.

**Relevance:** Ensure the summary and conclusions are directly related to the topic.

**Inviting Questions/Discussion:**

**Openness:** Be welcoming to questions and discussions from the audience.

**Encouragement:** Actively encourage audience interaction.

**Responsiveness:** Address audience questions and feedback effectively and clearly.