

# Informática nas Organizações

---

Natércia Menezes – [ncm@isep.ipp.pt](mailto:ncm@isep.ipp.pt)

ISEP, 2022-2023

## Diagramas de componentes e de deployment

### Component and deployment diagrams

---

#### **O que é um Diagrama de Componentes?**

Os diagramas de componentes são usados para visualizar a organização dos componentes do sistema e os relacionamentos de dependência entre eles.

Eles fornecem uma visão de alto nível dos componentes dentro de um sistema.

Os componentes podem ser um componente de software, como uma base de dados ou interface de utilizador; ou um componente de hardware, como um circuito, microchip ou dispositivo; ou uma unidade de negócios, como fornecedor, folha de pagamento ou envio.



## Diagramas de componentes e de deployment

### Component and deployment diagrams

---

#### What is a Component Diagram?

Component diagrams are used to visualize the organization of system components and the dependency relationships between them. They provide a high-level view of the components within a system. The components can be a software component, such as a database or user interface; or a hardware component, such as a circuit, microchip or device; or a business unit, such as vendor, payroll, or shipping.

## Diagramas de componentes component diagrams

---

- São utilizados no desenvolvimento baseado em componentes para descrever sistemas com arquitetura orientada a serviços.
- Mostrar a estrutura do próprio código.
- Pode ser usado para focar na relação entre componentes enquanto esconde detalhes de especificação.
- Ajudar a comunicar e explicar as funções do sistema que está a ser construído às partes interessadas.

## Diagramas de componentes component diagrams

---

- They are used in component-based development to describe systems with a service-oriented architecture.
- Show the structure of the code itself.
- It can be used to focus on the relationship between components while hiding specification details.
- Help communicate and explain the functions of the system being built to stakeholders.

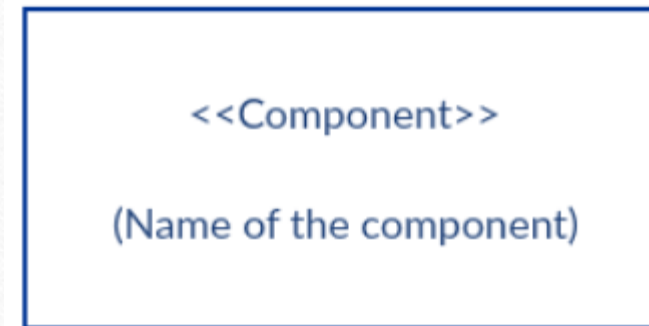


## Símbolos do Diagrama de Componentes – Componente

Existem três maneiras de usar o símbolo do componente

---

Retângulo com o estereótipo do componente (o texto <<componente>>). O estereótipo do componente é normalmente utilizado acima do nome do componente para evitar confundir a forma com um ícone de classe.



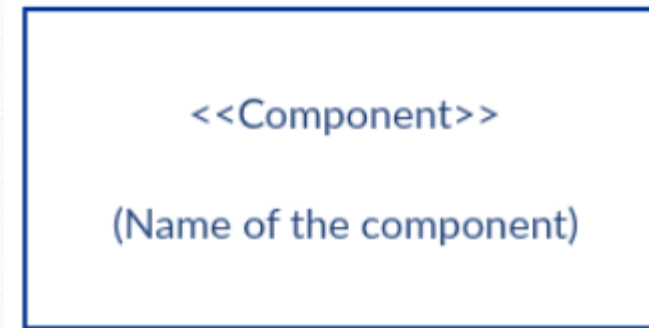
## Component Diagram Symbols – Component

There are three ways to use the component symbol

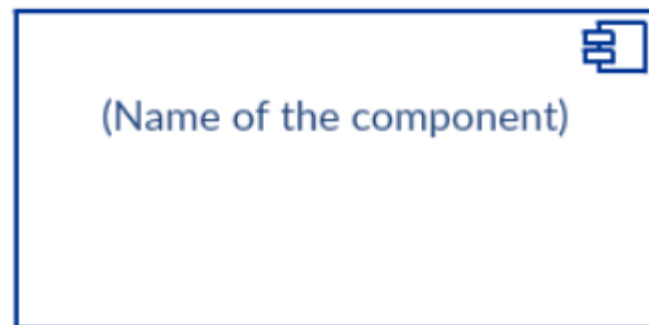
---

Rectangle with the component stereotype (the text <<component>>).

The component stereotype is normally used above the component name to avoid confusing the shape with a class icon.

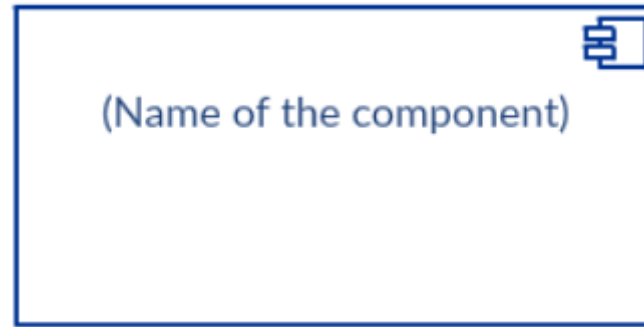


Retângulo com o ícone do componente no canto superior direito e o nome do componente.

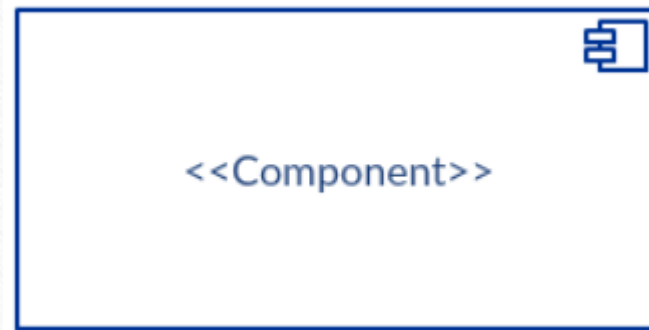




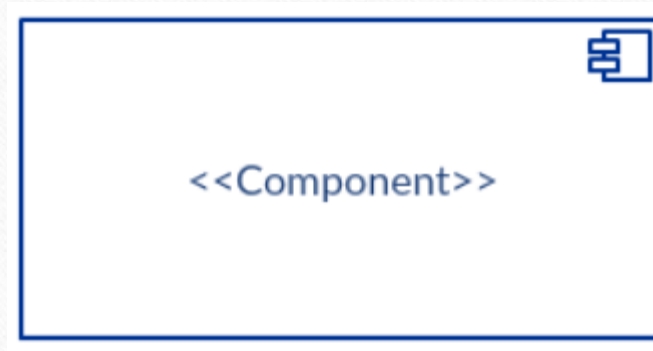
Rectangle with the component icon in the upper right corner and the component name.



Retângulo com o ícone do componente e o estereótipo do componente.

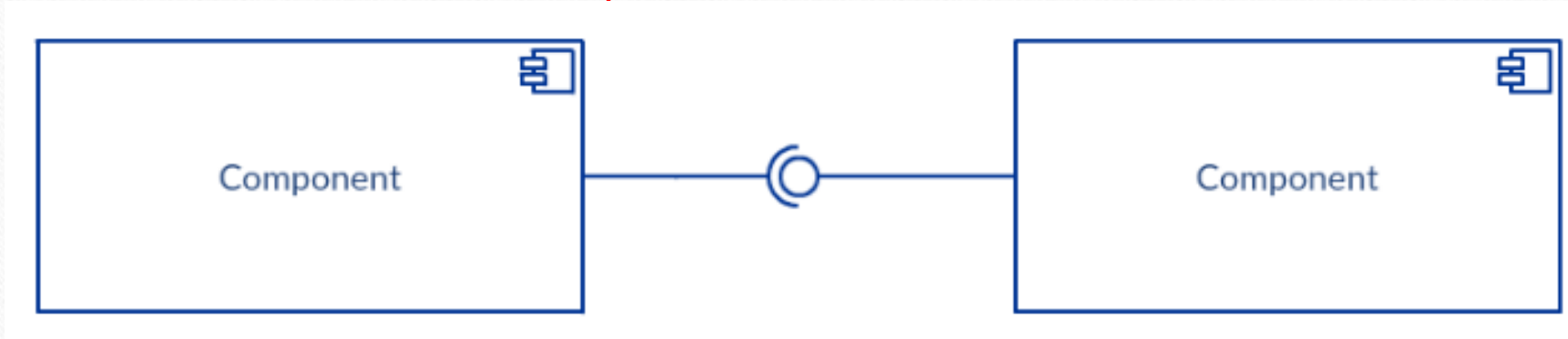


Rectangle with component icon and component stereotype.



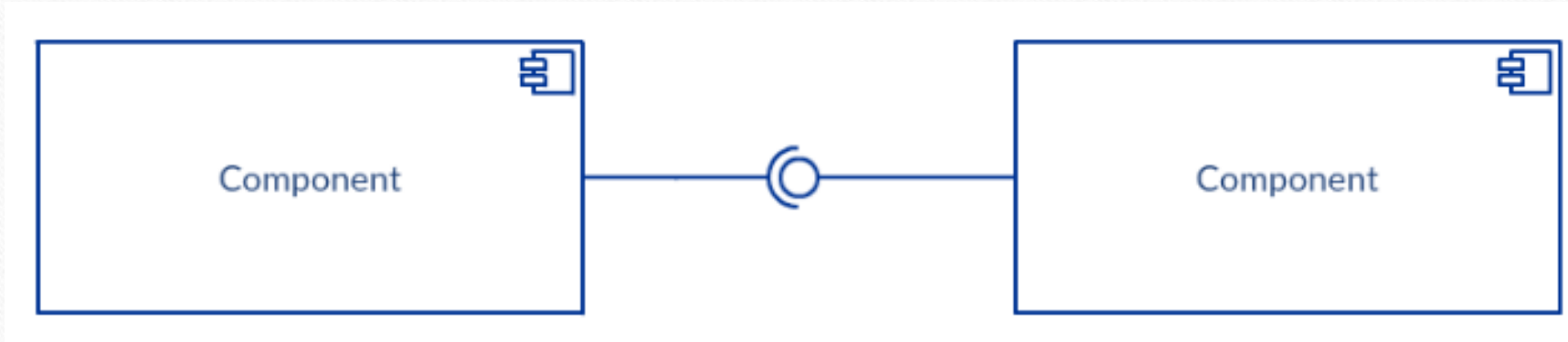


## Interface Fornecida e uma Interface Requerida Interface Provided e a Interface Requerida



As interfaces nos diagramas de componentes mostram como os componentes são interligados e interagem entre si. O conector de assembleia permite vincular a interface necessária do componente (representado com um semicírculo e uma linha sólida) com a interface fornecida (representada com um círculo e uma linha sólida) de outro componente. Isso mostra que um componente está fornecendo o serviço que o outro exige.

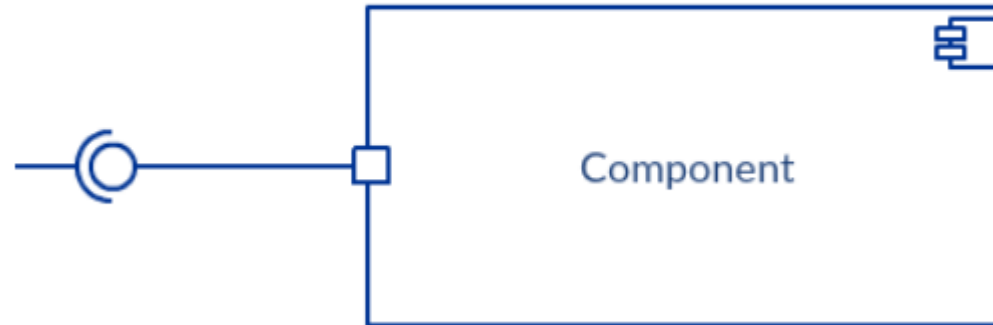
## Interface Provided e a Interface Requerida



Interfaces in component diagrams show how components are interconnected and interact with each other. The assembly connector allows you to link the required interface of the component (represented with a semicircle and a solid line) with the provided interface (represented with a circle and a solid line) of another component. This shows that one component is providing the service that the other requires.

# Porta

Porta (representada pelo pequeno quadrado no final de uma interface necessária ou interface fornecida) é usada quando o componente delega as interfaces a uma classe interna.

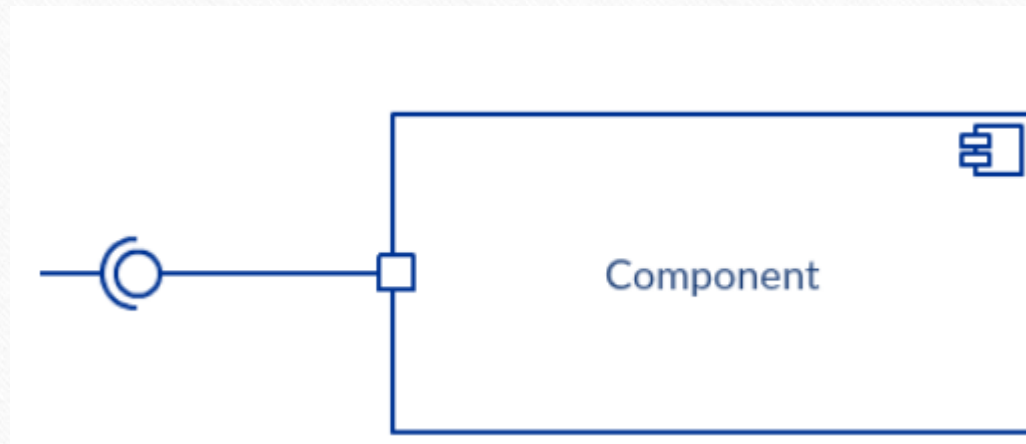




# Port

---

Port (represented by the small square at the end of a required or provided interface) is used when the component delegates interfaces to an inner class.



# Dependências

---

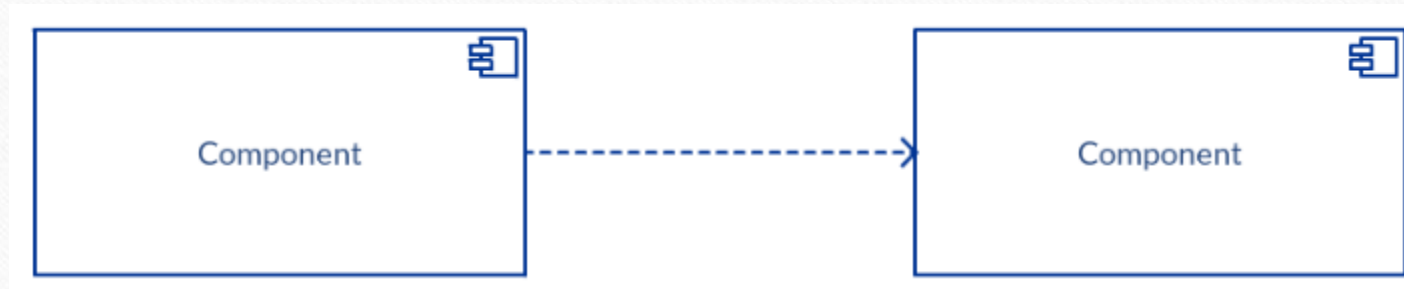
Também é possível usar uma seta de dependência para mostrar a relação entre dois componentes.



# Dependency

---

You can also use a dependency arrow to show the relationship between two components.





# Como desenhar um diagrama de componentes

## How to draw a component diagram

---

Pode usar um diagrama de componentes quando quiser representar o sistema como componentes e quiser mostrar as suas inter-relações através de interfaces.

Ajudam a ter uma ideia da implementação do sistema.

You can use a component diagram when you want to represent the system as components and you want to show their interrelationships through interfaces. They help to get an idea of the implementation of the system.

## **Passos que podem seguir ao desenhar um diagrama de componentes:**

**Passo 1:** Descubra a finalidade do diagrama e identifique os artefatos como ficheiros, documentos, etc no seu sistema ou aplicação que precisa representar o seu diagrama.

**Passo 2:** Conforme descobre as relações entre os elementos identificados anteriormente, crie um layout mental do seu diagrama de componentes.

**Passo 3:** Enquanto desenha o diagrama, adicione os componentes primeiro, agrupando-os dentro de outros componentes como achar melhor.

**Passo 4:** O próximo passo é adicionar outros elementos, como interfaces, classes, objetos, dependências, etc ao diagrama de componentes e completá-lo.

**Passo 5:** Pode anexar notas sobre diferentes partes do diagrama de componentes para esclarecer certos detalhes a outros.



## **Steps you can follow when drawing a component diagram:**

**Step 1:** Discover the purpose of the diagram and identify artifacts such as files, documents, etc. in your system or application that need to represent your diagram.

**Step 2:** As you discover the relationships between the elements identified earlier, create a mental layout of your component diagram.

**Step 3:** While drawing the diagram, add the components first, grouping them inside other components as you see fit.

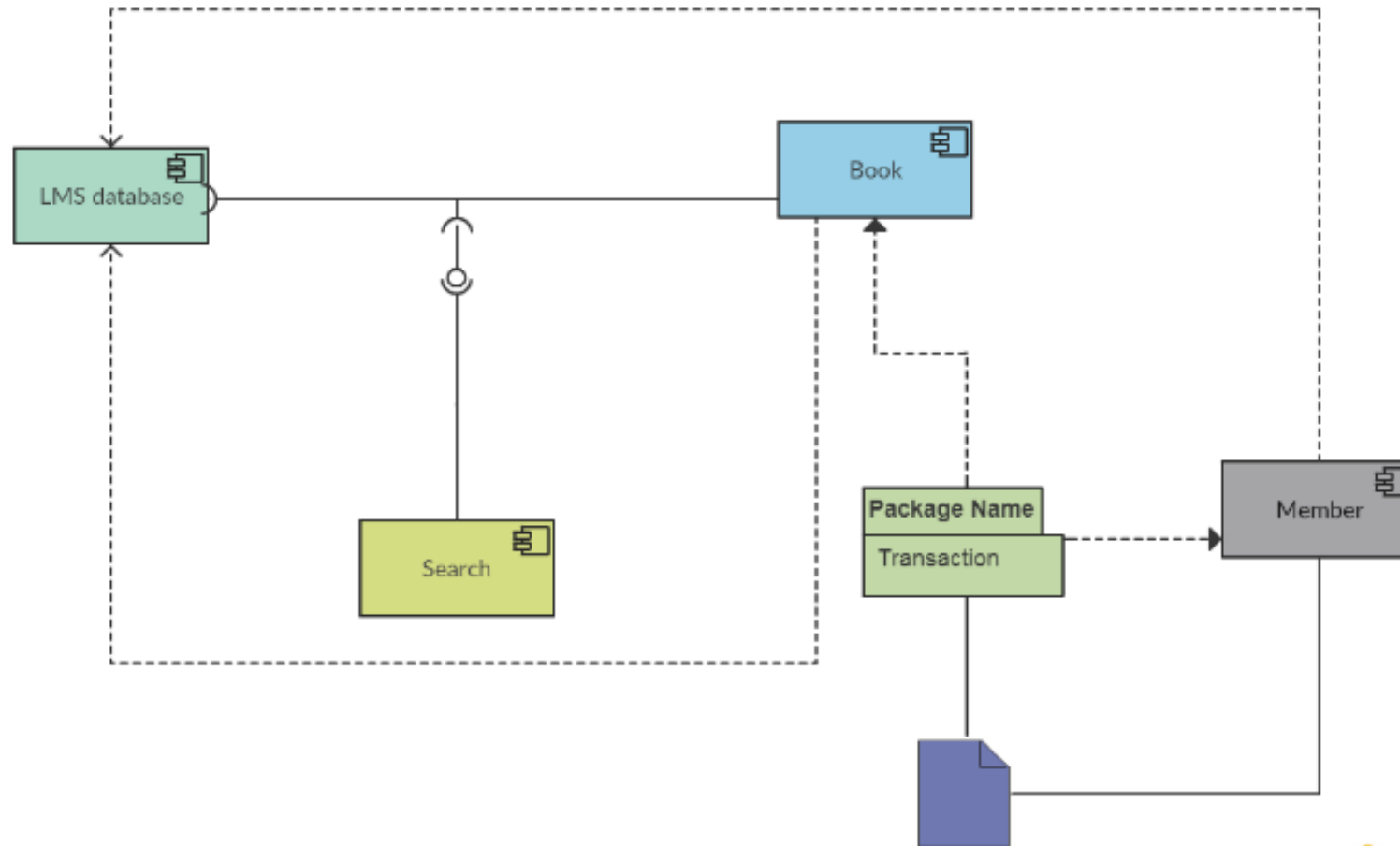
**Step 4:** The next step is to add other elements such as interfaces, classes, objects, dependencies, etc. to the component diagram and complete it.

**Step 5:** You can attach notes about different parts of the component diagram to clarify certain details to others.



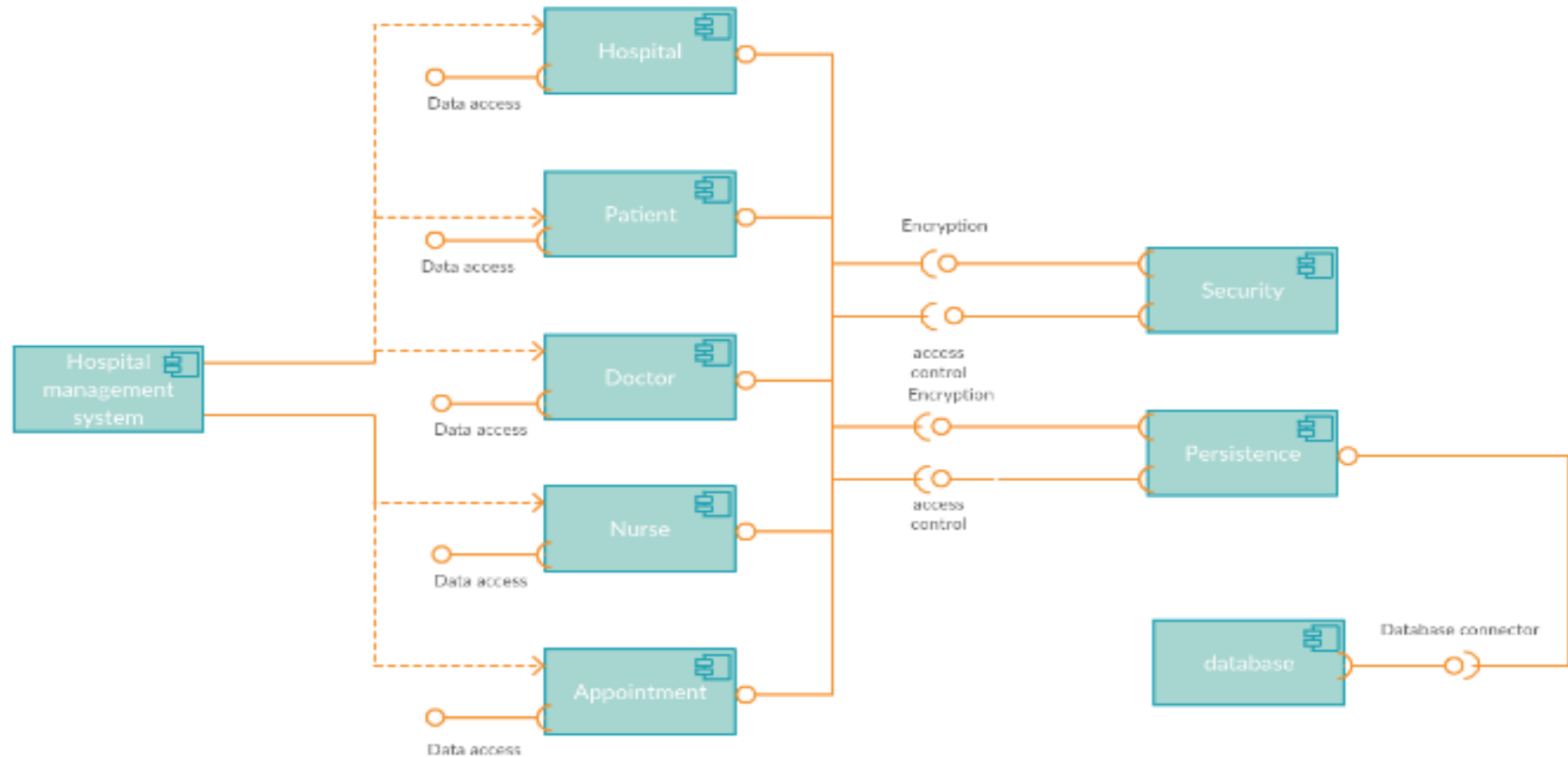
## Diagrama de Componentes para o Sistema de Gestão de Bibliotecas

### Component Diagram for the Library Management System



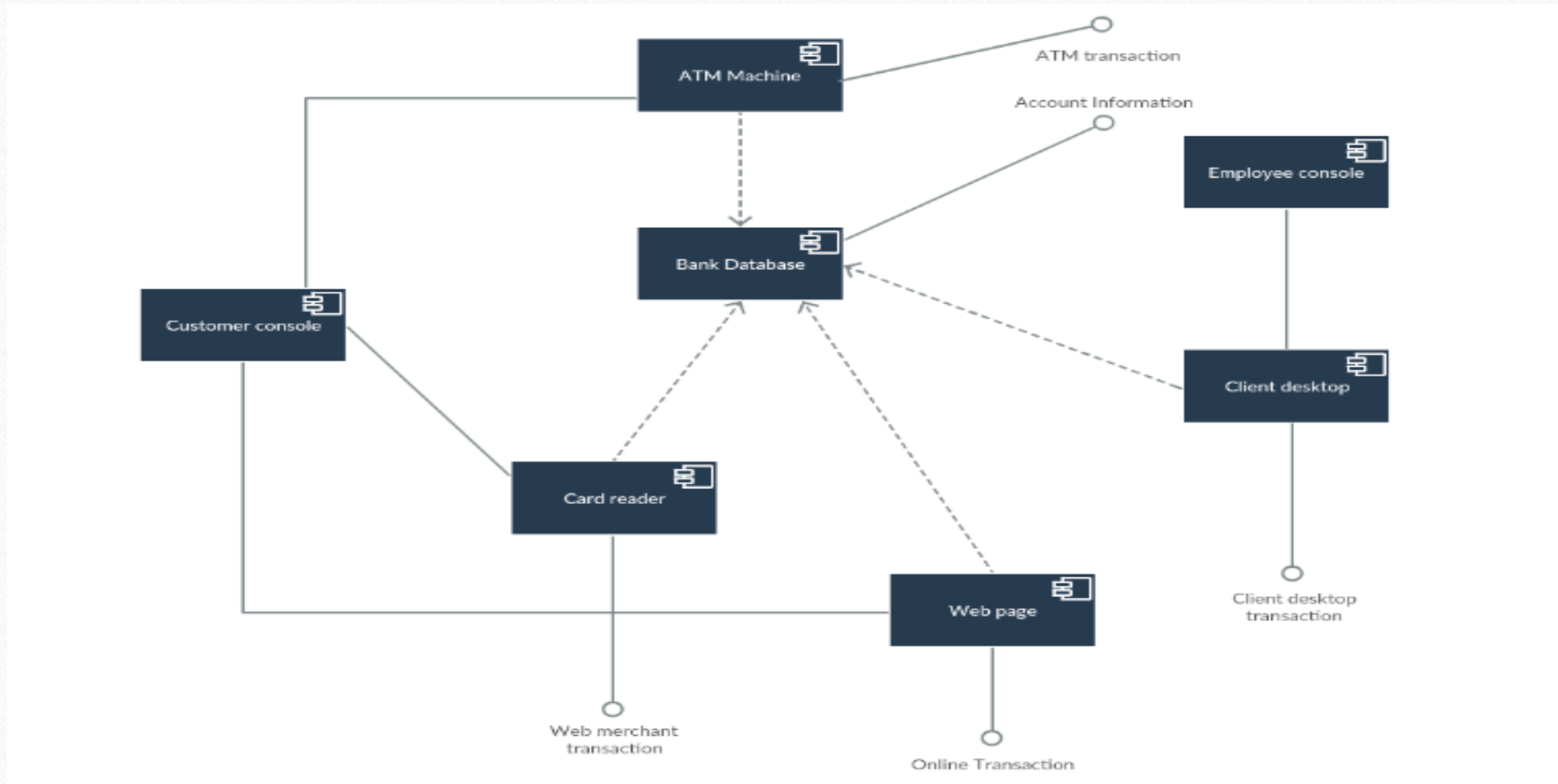
## Diagrama de Componentes para Sistema de Gestão Hospitalar

### Component Diagram of Hospital Management System



## Diagrama de componentes para ATM

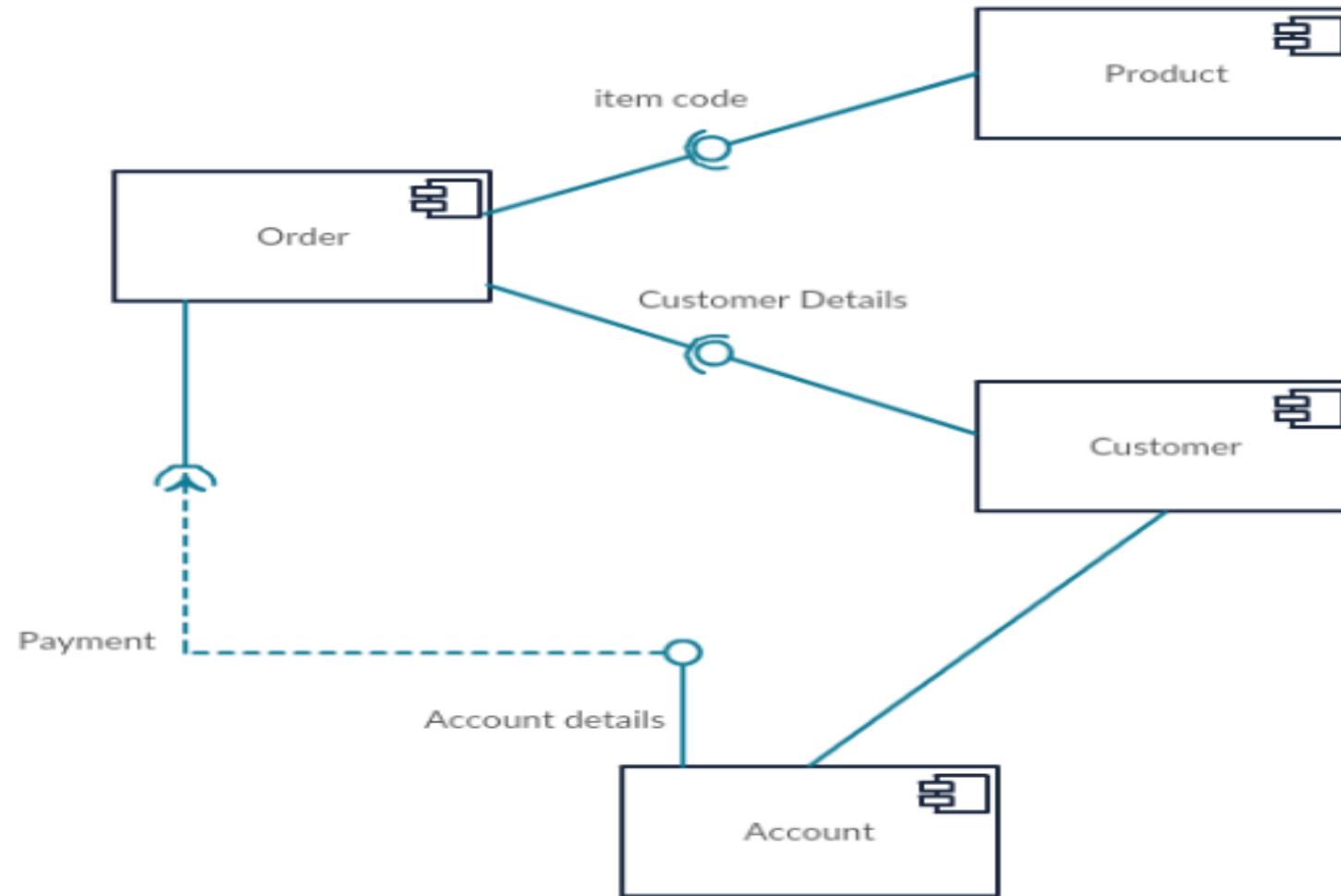
### Component diagram for ATM





## Diagrama de Componentes para Sistema de Compras Online

### Component Diagram for Online Shopping System



## Diagrama de componentes para Sistema de Gestão de stokes

### Component diagram for Stock Management System

