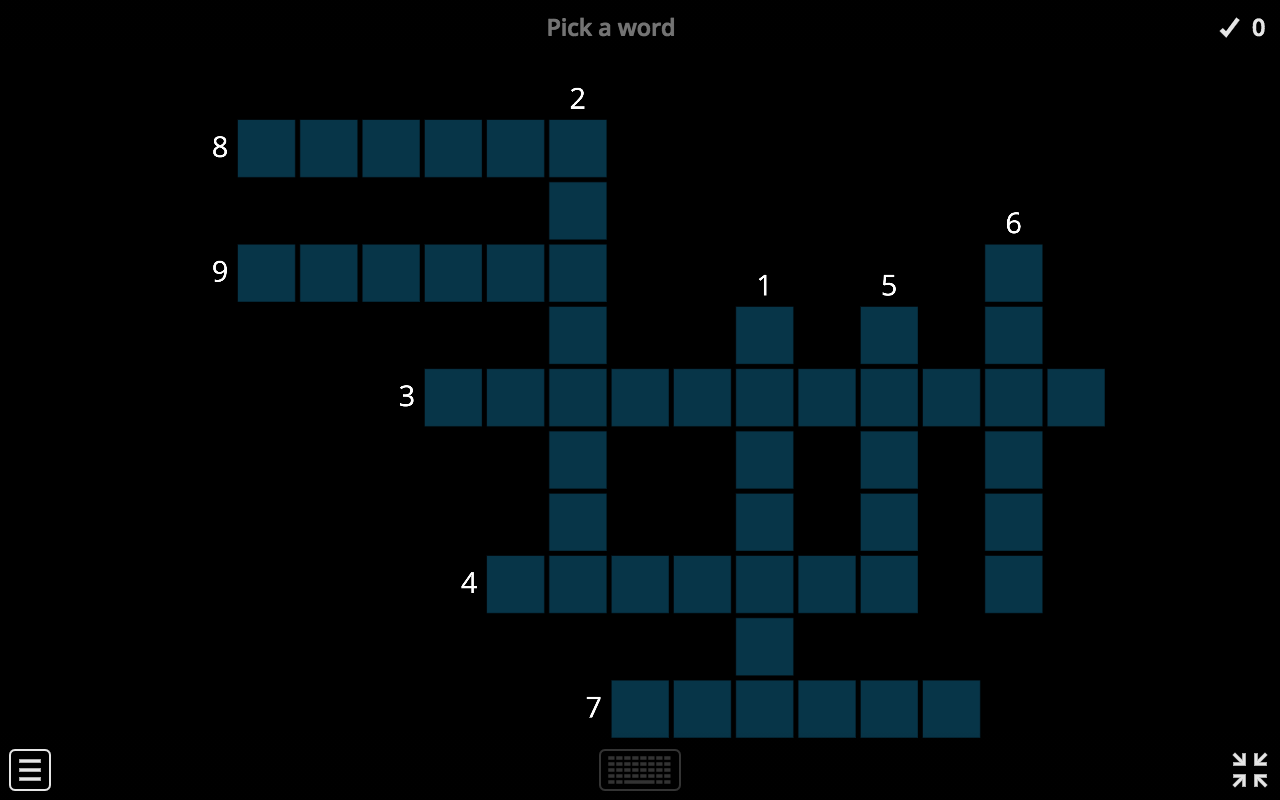


**MVC ARCHITECTURE**

**Worksheet 2.1**

**Relate the words below and clues to complete the crossword.**

* **Handle:resolver un problema,situacion/empuñadura**
* **Manage:gestionar,administrar**
* **Pattern:patron ,modelo,secuencia**
* **Output:salida,resultado**
* **Inputs:entrada,aportes**
* **Retrieve´:recuperar,rescatar,Volver a traer**
* **Render:renderizar,generar imagen fotorealista**
* **Request:peticion,requerimiento,solicitud**
* **Maintaining:mantenimiento**

****

1. \_\_\_\_\_pattern\_\_\_\_\_ :It is a repeatable solution to a software engineering problem.
2. \_\_\_retrieve\_\_\_\_\_\_\_ :It is the process of searching for, locating, and returning data.
3. \_\_\_\_mainteining\_\_\_\_\_\_ :A computer system that is essential for the performance and lifespan of a computer.
4. \_\_\_\_request\_\_\_\_\_\_ :The state of being asked for or wanted.
5. \_\_\_input\_\_\_\_\_\_\_ :Any information or data sent to a computer for processing.
6. \_\_\_\_handle\_\_\_\_\_\_ :It is an abstract reference to a resource that is used when application software references blocks of memory or objects managed by another system.
7. \_\_\_manage\_\_\_\_\_\_\_ :It Is the process of managing, monitoring and optimizing a computer system for performance.
8. \_\_\_render\_\_\_\_\_\_\_ :Refers to the act of drawing the modelled objects on screen.
9. \_\_output\_\_\_\_\_\_\_\_ :Data generated by a computer.

**Worksheet 2.2**

**MVC Framework**

**What is MVC Framework?**

The Model-View-Controller (MVC) framework is an architectural **pattern** that separates an application into three principal components Model, View, and Controller. Each architecture component manages a specific development aspect of an application.

Three important MVC the components are:

* Model: It includes all the data and its related logic.
* View: It presents data to the user or handles user interaction.
* Controller: An interface between Model and View components.

**View**

A View is that part of the application that represents the presentation of data. Views are created by the data collected from the model data. A View requests the Model to give information so that it presents the **output** presentation to the user.

**Controller**

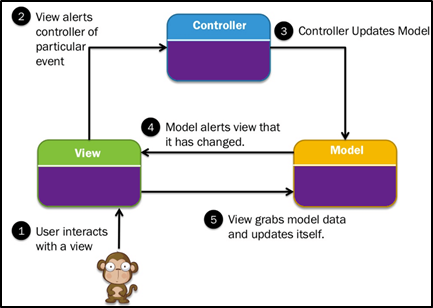
The Controller is that part of the application that **manages** the user interaction. The Controller interprets the mouse and keyboard

**inputs** from the user, informing Model and View to change as appropriate. A Controller sends commands to the model to update its state (E.g., Saving a specific document). The Controller also sends commands to its associated View to change the View's presentation (For example, scrolling a particular document).

**Model**

The Model component stores data and its related logic. For example, a Controller object will **retrieve** the customer info from the database. It manipulates data and sends back to the database or uses it to **render** the same data. It responds to the **request** from the View and also responds to instructions from the Controller to update itself. It is also the lowest level of the pattern which is responsible for **maintaining** data.

In the diagram below, you can see the visual representation of the MVC architecture.

****

*MVC Architecture Diagram*

**MVC Example**

Let's see a Model View Controller example from daily life:

****

Let's imagine you go to a restaurant. You go there, sit at the table and wait for the waiter to come. The waiter comes to you, and you order the food. The waiter doesn't know who you are and he just writes down the details of your food order. Then, the waiter goes to the kitchen. In the kitchen, the waiter does not prepare your food - the cook prepares your food. The waiter gives your order to the cook along with your table number. The cook then prepares food for you. He uses ingredients to cook the food. Let's imagine that you ordered a vegetable sandwich. Then he needs bread, tomato, potato, capsicum, onion, bit, cheese, etc. which he takes from the refrigerator. Finally, the cook hands over the food to the waiter. Now it is the job of the waiter to take this food outside the kitchen and serve it to you.

In this MVC example,

* Client = View
* Waiter = Controller
* Cook = Model
* Refrigerator = Data

**Advantages of MVC: Key Benefits**

What are the major benefits of using MVC architecture? First of all, it has code maintenance that is easy to extend and grow. Second, MVC Model component can be tested separately from the user and it offers the best support for [test-driven development](https://www.guru99.com/test-driven-development.html). In addition to that, all classes and objects are independent from each other so you can test them separately. Third, it helps you to avoid complexity by dividing an application into the three units: model, view, and controller. It also works well for Web apps which are supported by large teams of web designers and developers.

**Disadvantages of using MVC**

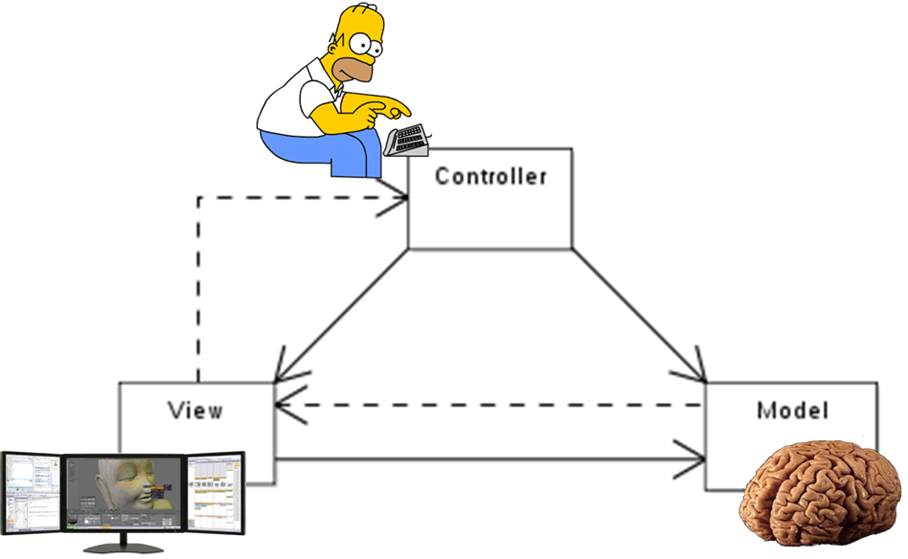
However, there are some disadvantages in the MVC model. First, it is difficult to read, change, to unit test, and reuse this model. Second, there is a need for multiple programmers to conduct parallel programming. Third, the developers need to have good knowledge of multiple technologies. Finally, the programmers need to maintain many codes in Controller.

**Worksheet 2.3**

**WRAP-UP**

**Read the text and after that, you will paraphrase the main ideas of each of the following sections:**

* What is MVC Framework? A MVC framework is a desing patter of OOP based in leyered architecture in which divide the code in model,view and controller
* View is the visual part interactive with the user and receive the request connect with the controller and show the output
* Controller this is the intermediary with the view and model this execute the logical process about resolve the user request
* Model this is the classes or also are the database tables and follow receive and save the data if this program have database and send this request to the controller and view to the user
* Advantages of MVC: Key Benefits: avoid redundancy,easy maintance,better order of code,responsabilities independiently
* Disadvantages of using MVC: sometimes is confused with the order, and is difficult to execute unit test

****

**Worksheet 2.4**

**Ask yourself the following questions:**

1. ¿Entiendo que es MVC architecture?

Yes **😃** Maybe **😐** No **😟**

1. ¿Entiendo de qué componentes consiste MVC?

Yes **😃** Maybe **😐** No **😟**

1. ¿Entiendo las ventajas y desventajas de MVC?

Yes **😃** Maybe **😐** No **😟**

1. ¿Entiendo en qué consiste la estrategia de Paraphrasing?

Yes **😃** Maybe **😐** No **😟**

1. ¿La estrategia de Paraphrasing me ayuda a entender mejor un texto.?

Yes **😃** Maybe **😐** No **😟**