**RESUMEN ARQUITECTURA EN SOFTWARE**

**Arquitectura**

MVC – Modelo vista controlador

Es un patrón de arquitectura de software que separa la implementación de una API en 3 capas:

* Datos y lógica de negocio (frontend y backend).
* Módulo encargado de gestionar los eventos y comunicaciones con servicios

**Control de versiones, GIT y github**

Control de versiones: gestión de los diversos cambios que se realizan

Principales características de GIT:

Branching model:

* Es una convención que se utiliza en la gestión y control de versiones de un proyecto, con el objetivo de simplificar la gestión de cambios

**INTRODUCCION A GO**

Lenguaje de programación compilado.

No usa ; al final de las líneas.

Características:

* Tipado estático
* Garbage collection
* Administración segura de la memoria
* Concurrencia: más asociado al diseño de software
* Paralelismo: funciones se ejecuten al mismo tiempo - hardware
* Velocidad

func main(){ } - Inicia el programa.

import - Uso librería externa.

fmt (librería externa reemplaza al std).

Ejemplo de programa:

func main() {

var a int = 10 // a := 10 (se puede declarar de ambas formas pero si declaro de la segunda forma no puedo cambiar el tipo de variable cuando le asigno un valor a ‘a’)

fmt.Println(a)

}

Receivers

Permiten implementar métodos que formen parte de una estructura. De esta manera, yo puedo crear una estructura y luego definir para esa estructura una función mediante un receiver para luego invocara.

