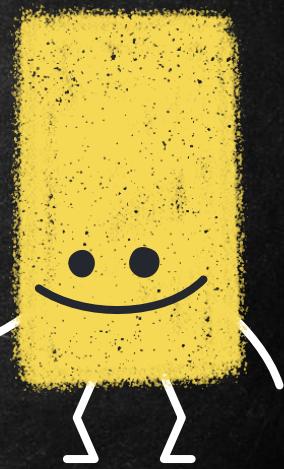
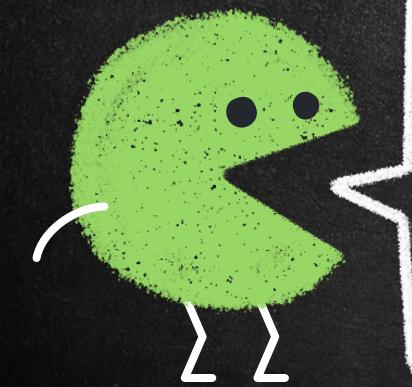
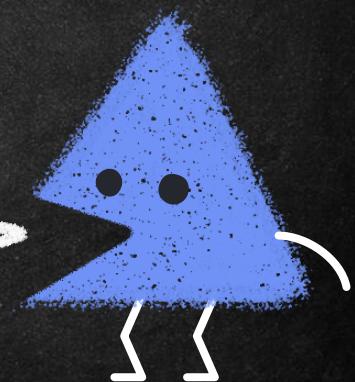


# LABORATORIO ED1



## SOBRE MÍ:

- Karen Liska
- Ing. En Informática y Sistemas
- Mgtr. Seguridad Informática
- Experiencia en desarrollo de software
- Experiencia en calidad de software
- Karenliska@Gmail.com



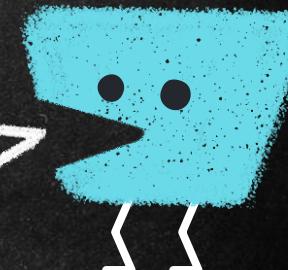
# PILARES DEL CURSO

## GIT

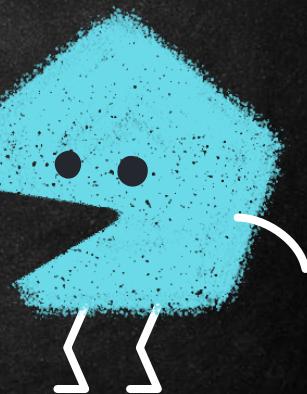
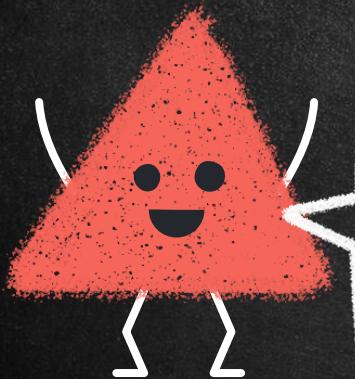
- Control de Versiones
- Trabajo colaborativo
- Trabajo organizado
- Trabajo a distancia

## MVC

- Desarrollo Web integral
- Mejores prácticas
- Estándar de desarrollo
- Programación orientada a objetos
- Grandes sistemas de información

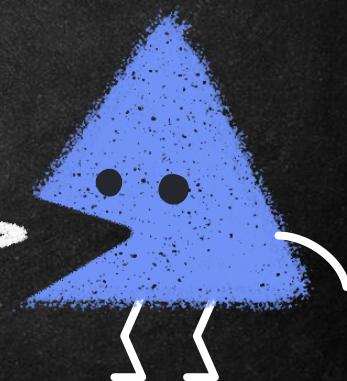


# MODO DE TRABAJO



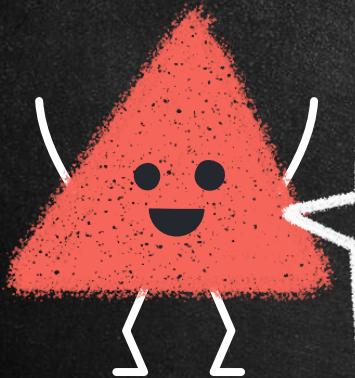
## MODO DE TRABAJO

- Laboratorios cada 2 semanas
- Grupos de 4 personas
- Entrega parcial Semanal
- Entrega parcial requisito de entrega final
- Todas las entregas son presenciales
- Nota proporcional a los aportes de cada miembro del equipo



# GIT

Control de versiones y trabajo  
colaborativo



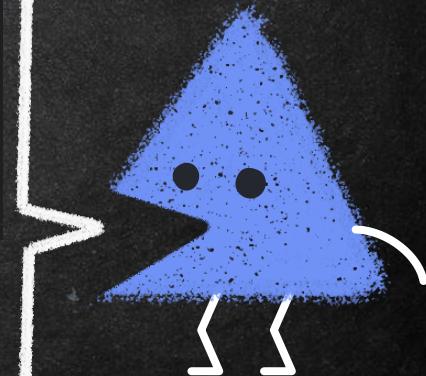
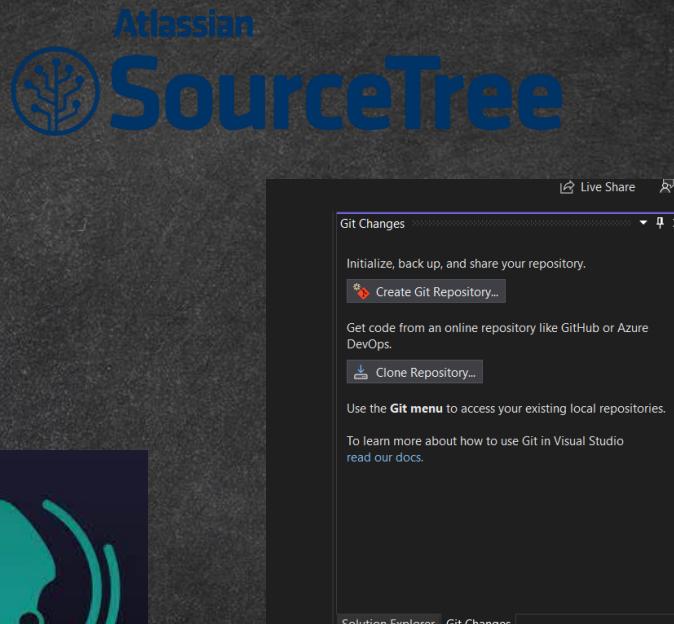
# POR QUÉ ES IMPORTANTE?

- De las tecnologías más utilizadas
- Hace posible el control de versiones
- Trabajo colaborativo
- Trabajo organizado
- Es requisito para trabajar
- Utilizado en sistemas grandes
- No solo se utiliza para programación

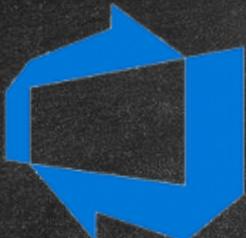


# HERRAMIENTAS

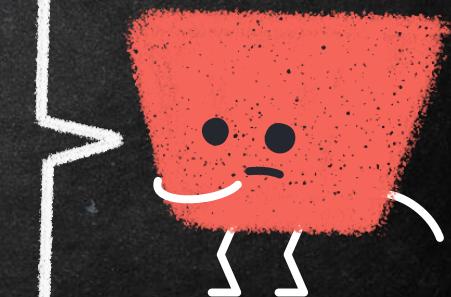
- GitBash
- SourceTree
- GitKraken
- Visual studio Git



# IMPLEMENTACIONES

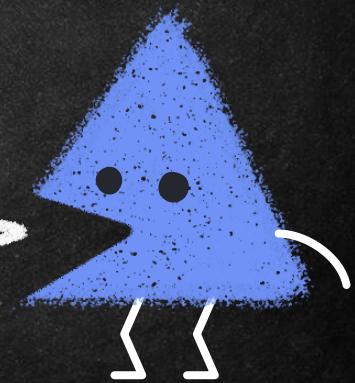


Azure DevOps

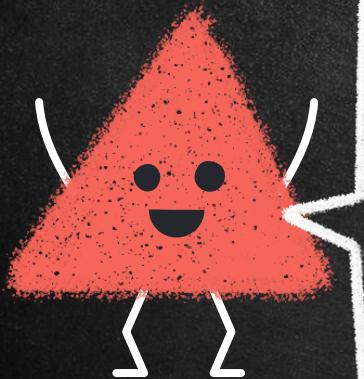


# CONCEPTOS IMPORTANTES

- Repositorio
- Repositorio Local
- Repositorio Remoto
- Rama
- Clonar
- Commit
- Fetch
- Push
- Merge
- Conflicts

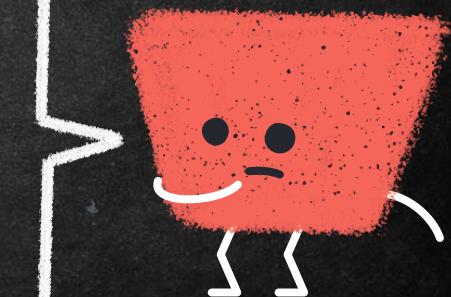
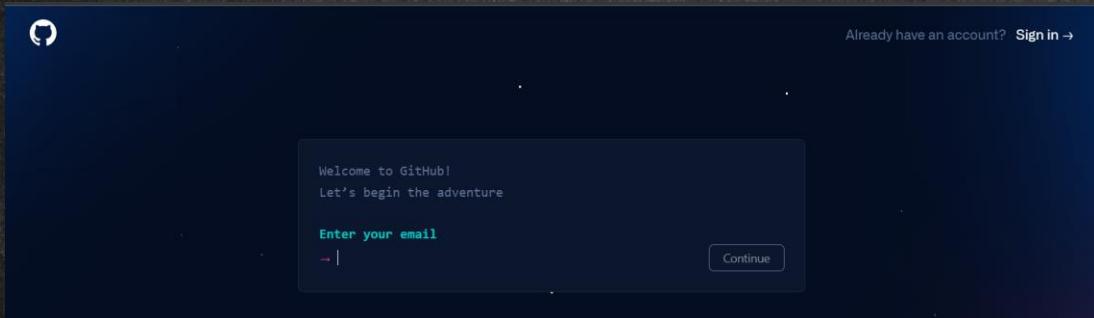


MANOS A LA OBRA

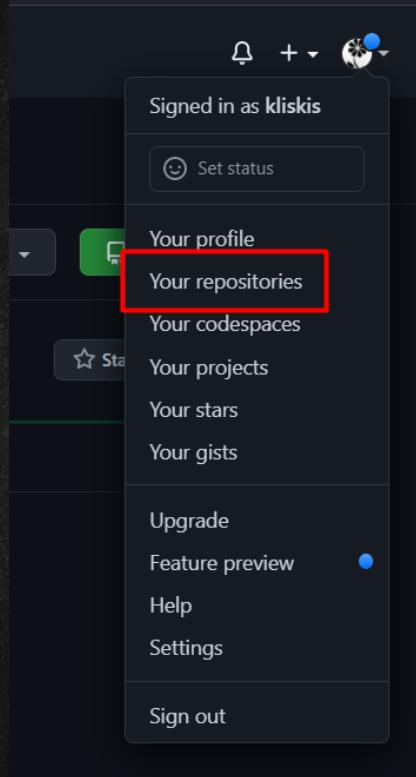


# CREAR UNA CUENTA

→ <https://github.com/>



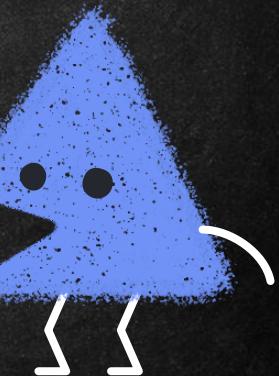
# CREAR UN REPOSITORIO



A screenshot of the GitHub web interface. The top navigation bar includes Overview, Repositories (1), Projects, Packages, Stars, Set status, and a New button. Below the navigation is a search bar with 'Find a repository...'. A repository card for 'container-expert-scratch' (Public) is displayed, showing it was forked from 'kontinu/container-expert-scratch'. It is written in Python and has 108 commits, last updated on Aug 24, 2020. The repository has 3 followers and 0 following. At the bottom of the card is an 'Edit profile' button. The background features a large, stylized blue arrow pointing to the right.

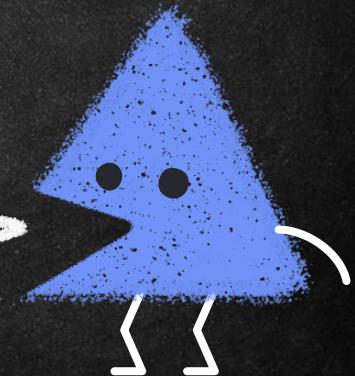
# FORMAS DE VINCULAR UN PROYECTO

- Clonar algo en blanco y copiar un proyecto en la carpeta
- Crear el repositorio local y luego publicarlo con todo su contenido



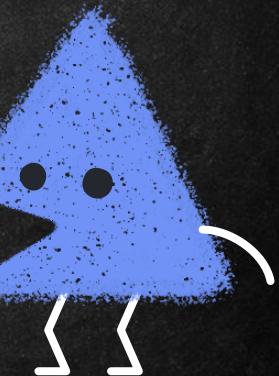
# COMANDOS CLAVE

- git config --global user.name
- git clone
- git add -A
- git status
- git commit -m “mensaje”
- git push



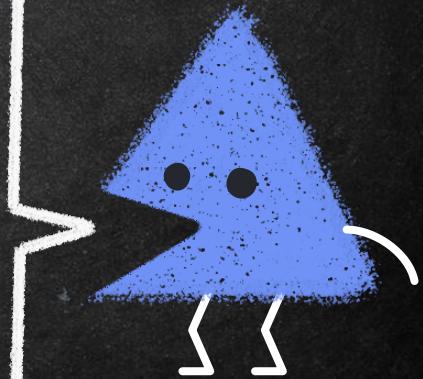
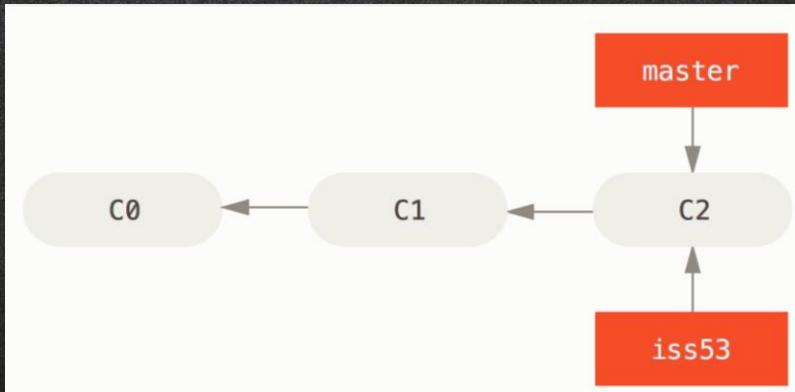
# BRANCHES

- Hablemos de master
- Feature branches
- git branch
- git checkout



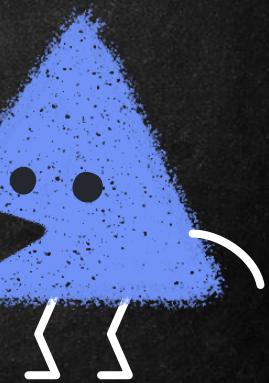
# MERGES

- Unificar ramas
- Añadir cambios de una rama a otra
- git checkout RamaDestino
- git merge RamaQueQueremosUnir

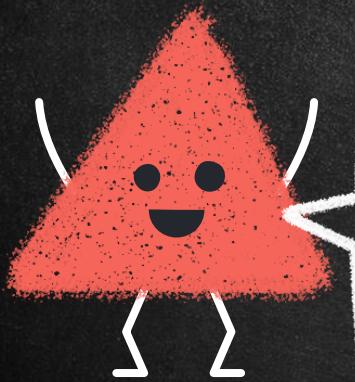


# CONFLICTOS Y RECOMENDACIONES

- Conflicto
- Forma de Resolver un conflicto



PASEMOS A VS



# CONECTARSE A GITHUB

## Clone a repository

Enter a Git repository URL

Repository location

`https://example.com/example.git <Required>`

Path

`C:\Users\karen\Source\Repos`

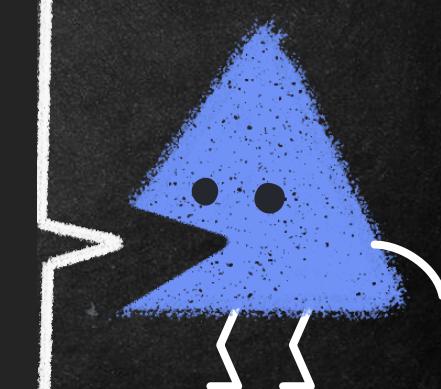
...

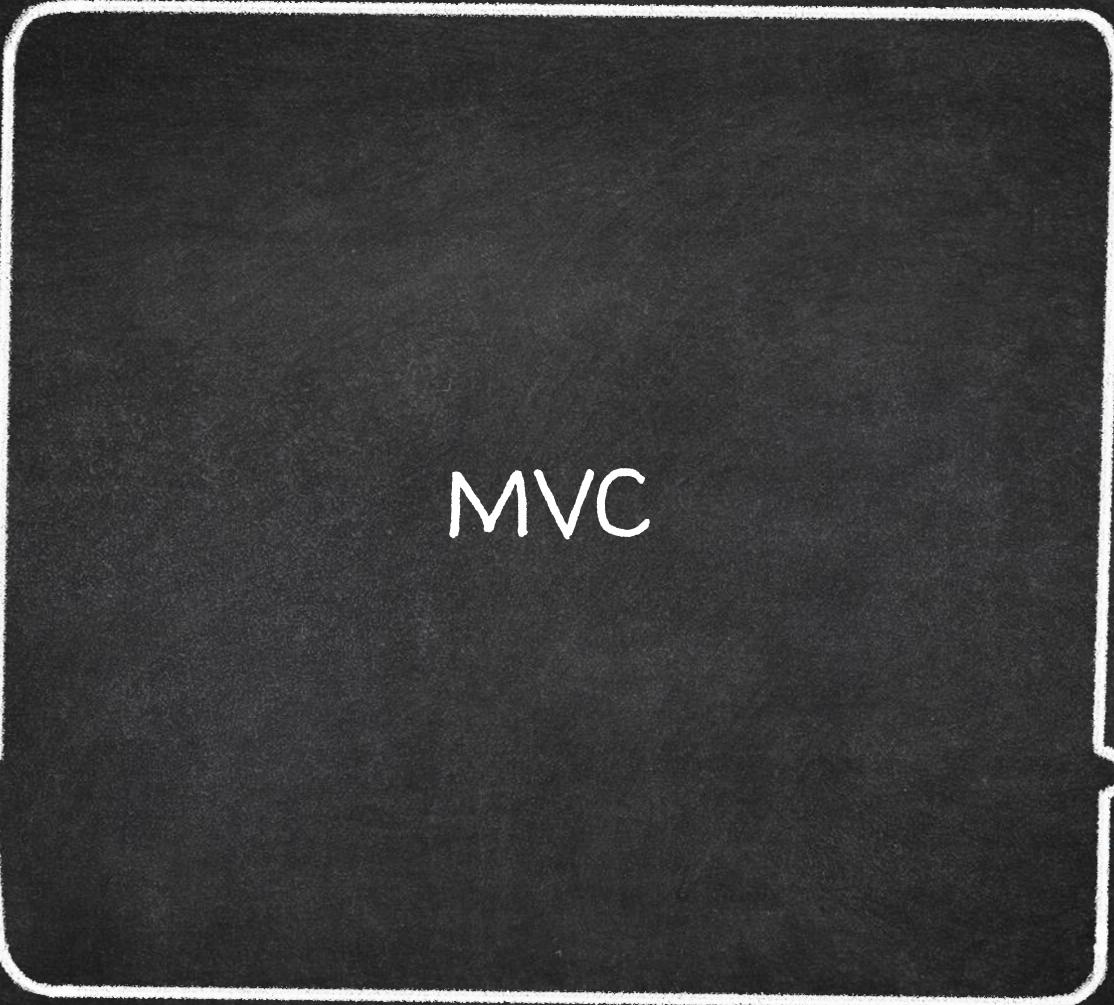
## Browse a repository

 Azure DevOps

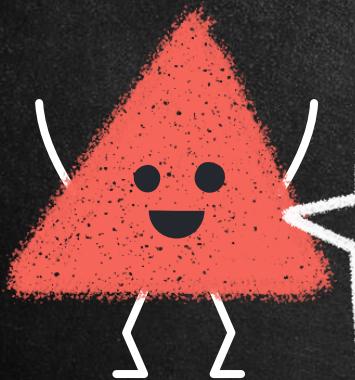
 GitHub

Clone





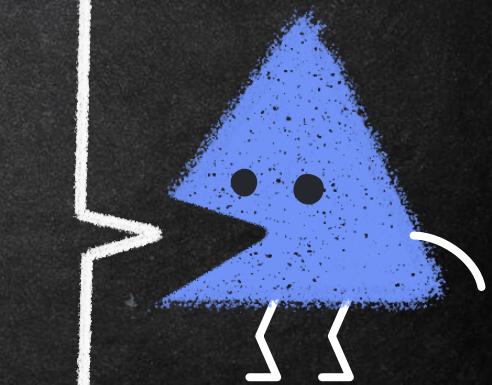
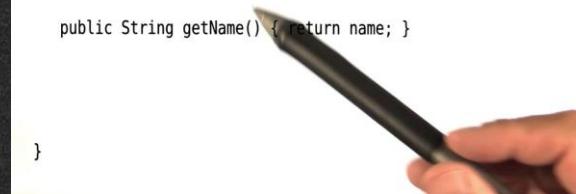
MVC



# MODELO – VISTA – CONTROLADOR

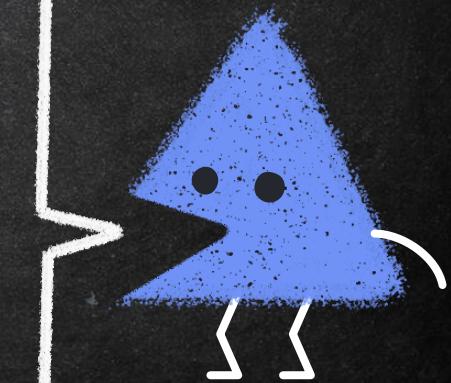
→ Modelo: van todas nuestras clases, todo lo que representa una entidad que poseen atributos y propiedades.

```
Properties  
public class Person  
{  
    private String name;  
  
    public String getName() { return name; }  
  
}
```



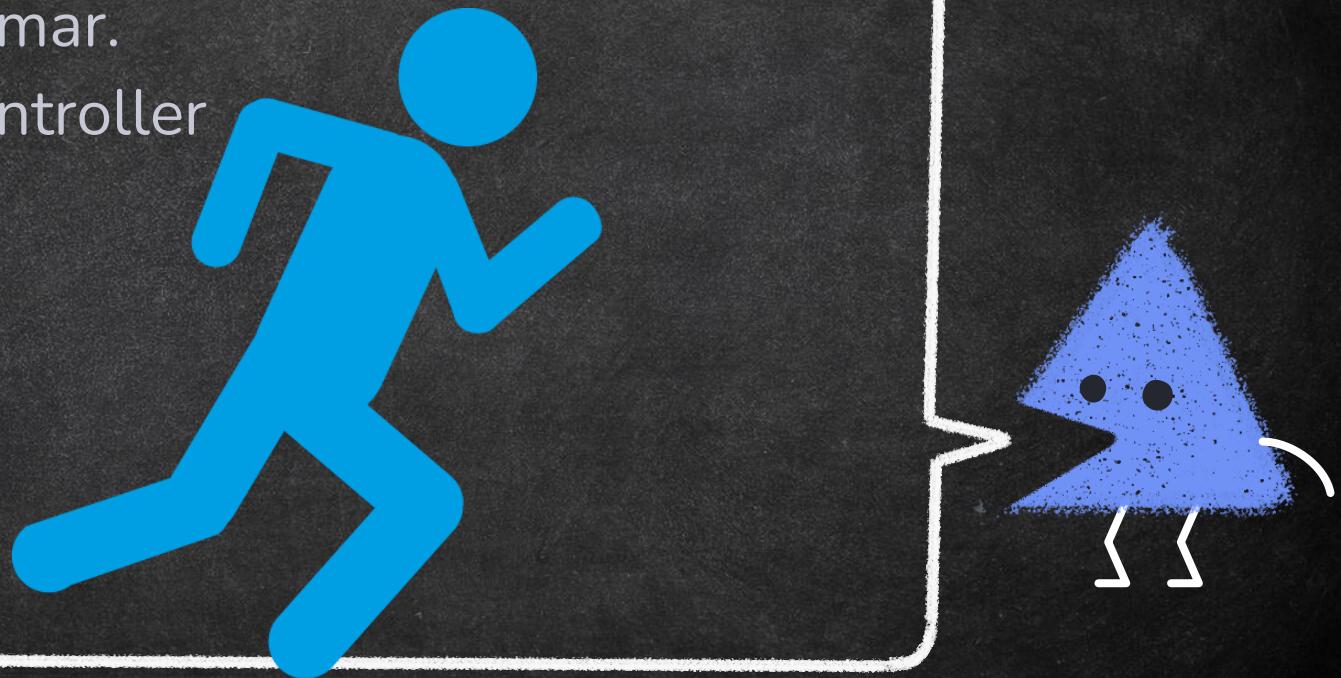
# MODELO – VISTA – CONTROLADOR

→ Vista: es la parte de la presentación, el cómo mostramos la información, puede ser super simple hasta super detallado.



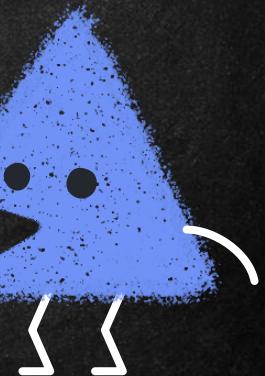
# MODELO – VISTA – CONTROLADOR

- Controlador: son las acciones que la vista manda a llamar.
- Nombre+controller



# STARTUP

- Inicio
- Estructura general de la aplicación
- Otras propiedades de arranque



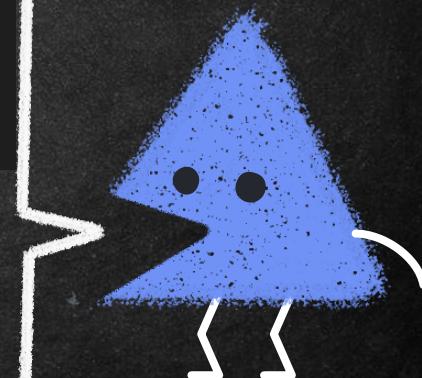
# DECORADORES

- Define atributos, tipos
- Cómo tratar una clase o método

```
namespace Api1.Controllers
{
    [Route("api/users")]
    [ApiController]
    public class UsersController : ControllerBase
    {
        private IUsers _usersList;
        public UsersController(IUsers users)
        {
            _usersList = users;
        }
    }
}
```

```
//API/Users
[HttpGet]
public IActionResult GetUsers()
{
    if (_usersList.GetUsers() == null)
        return BadRequest("No users found");
    return Ok(_usersList.GetUsers());
}
```

```
[TestClass]
public class UnitTest1
{
    [TestMethod]
    public void Test_AddMethod()
    {
        BasicMaths bm = new BasicMaths();
        double res = bm.Add(10, 10);
        Assert.AreEqual(res, 20);
    }
}
```



# TAREA

- Deberán crear su repositorio en Github
- Deberán investigar y elegir una de las herramientas a disposición para manejar GIT
- Deberán invitar al docente con el usuario kliskis a su repositorio privado

