

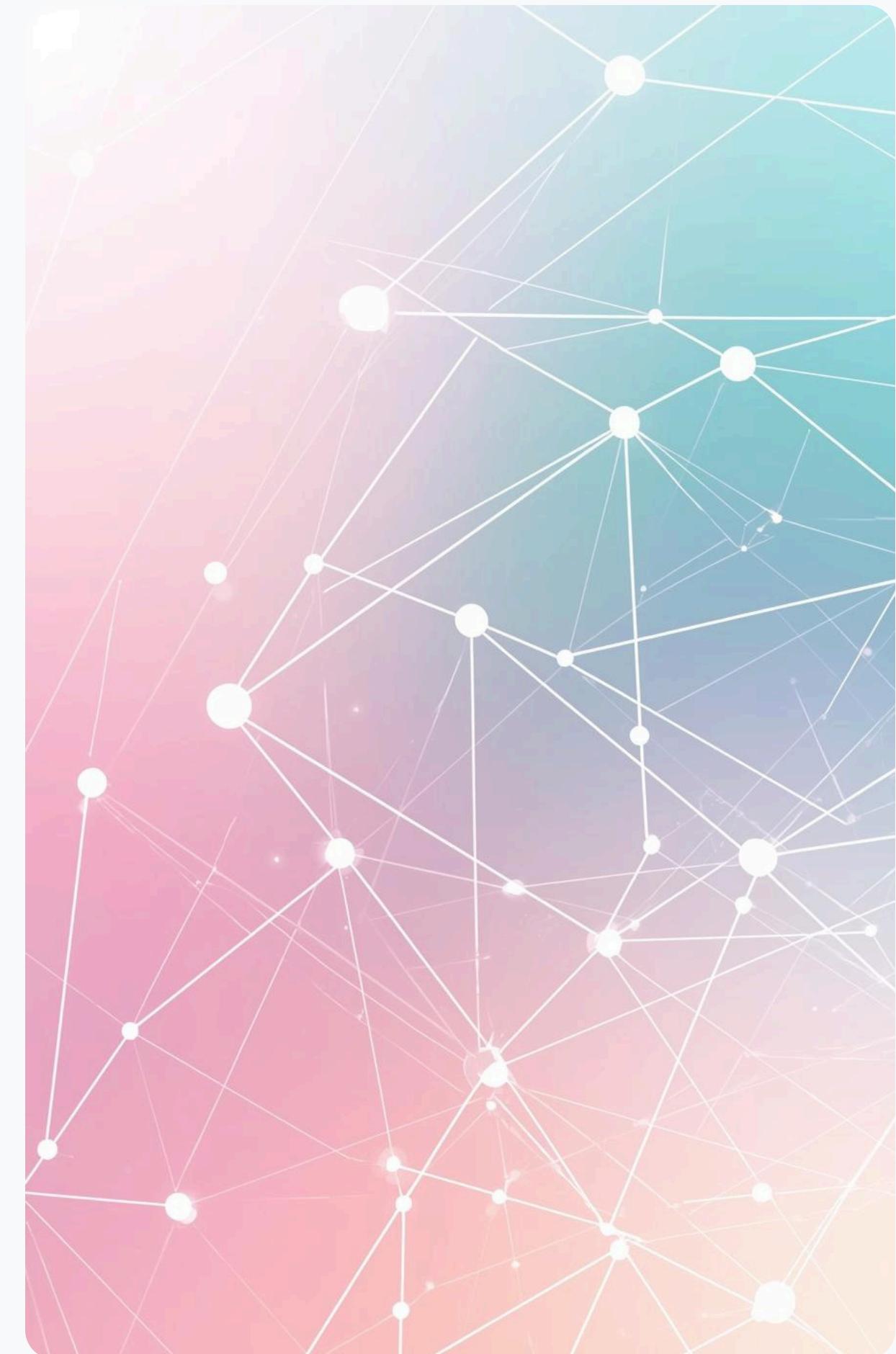
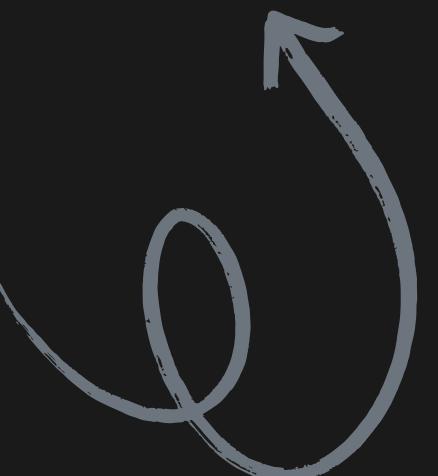
El impacto de las nuevas tecnologías

Proyecto Integrador

- Nombre: Ariel Ratto
- Materia: Logica de Programacion



Powered by
Arizona State University



Introducción a las nuevas tecnologías

- Las nuevas tecnologías forman parte de la vida cotidiana.
- Influyen en la comunicación, el aprendizaje y el entretenimiento.
- El software permite simular decisiones humanas de forma automatizada.



Problemas y Objetivos

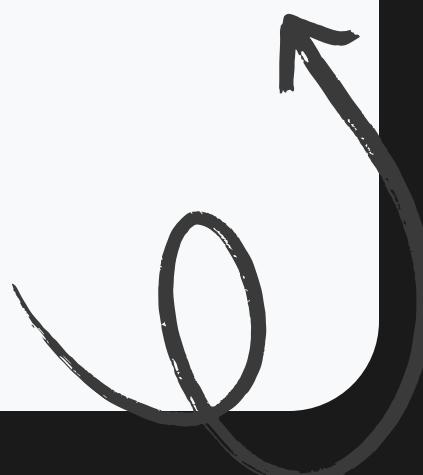
Afectación de las nuevas tecnologías en aspectos sociales

Problema:

- Muchas personas usan tecnología sin comprender cómo funciona.
- Incluso programas simples reflejan procesos tecnológicos complejos.

Objetivo:

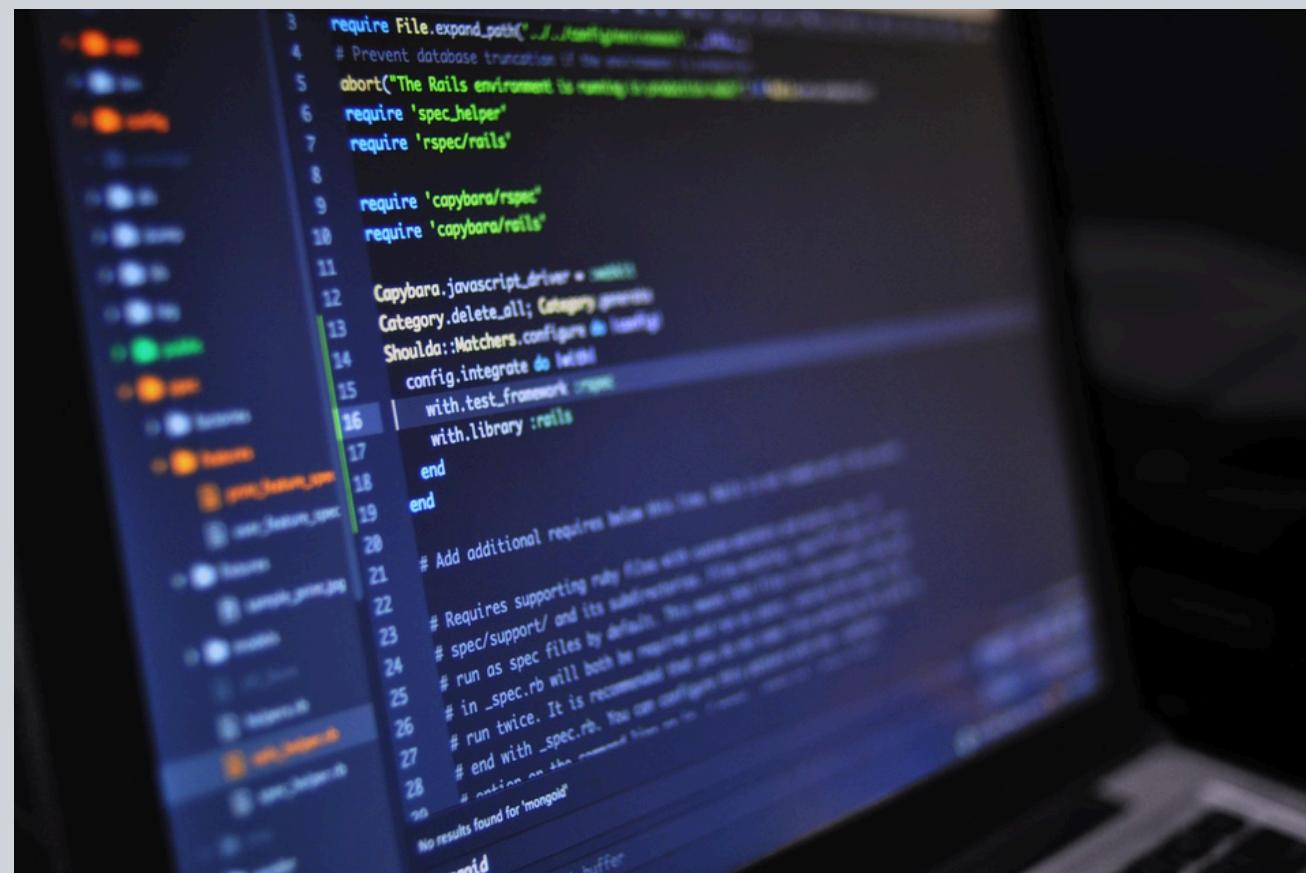
- Analizar el impacto de la tecnología en la sociedad.
- Desarrollar un software interactivo como ejemplo práctico.



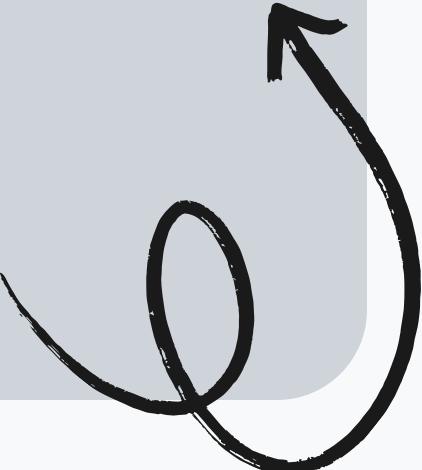
DESARROLLO DEL PROYECTO

Proceso de investigación y análisis

- Análisis del problema y planeación del software.
- Diseño del funcionamiento del programa.
- Desarrollo del código en Python.
- Pruebas y ajustes finales.



```
3 require File.expand_path('../config/environment', __FILE__)
4 # Prevent database truncation if the database needs cleaning
5 abort("The Rails environment is running in production mode!
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create!(name: "Electronics")
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with.test_framework :rspec
17     with.library :rails
18   end
19 end
20
21 # Add additional requires below this line if you need them
22
23 # Requires supporting files with custom matchers and helpers
24 # in ./support/ and its subdirectories. You can configure
25 # the location of these files by changing the
26 # --support flag in your RSpec configuration
27 # or in spec/spec.opts. It is recommended that you do
28 # this in your application's configuration file, 'config/
29 # application.rb', where it becomes available across all
30 # of your test files in that application.
31
32 # run as spec files by default. This means you don't
33 # have to run features/rspec/features.rb to run
34 # features/rspec/features/*.rb files. To change this, you
35 # can either
36 # a) set 'rspec --color' in your terminal
37 # b) set 'rspec --color=never' in your RSpec configuration
38 # c) set 'rspec --color=always' in your RSpec configuration
39
40 # run twice. It is recommended that you do
41 # this in your application's configuration file, 'config/
42 # application.rb', where it becomes available across all
43 # of your test files in that application.
44
45 # end with _spec.rb. You can configure
46 # this in your RSpec configuration
47 # or in spec/spec.opts. It is recommended that you do
48 # this in your application's configuration file, 'config/
49 # application.rb', where it becomes available across all
50 # of your test files in that application.
51
52 # option on the command line
53
54 # mongoid
55 # buffer
```



Desarrollo de un juego en Python

Título: Juego Piedra, Papel o Tijera

- Software desarrollado en Python.
- Interacción entre usuario y computadora.
- La computadora toma decisiones aleatorias.
- Simula procesos básicos de decisión.



FUNCIONAMIENTO DEL JUEGO

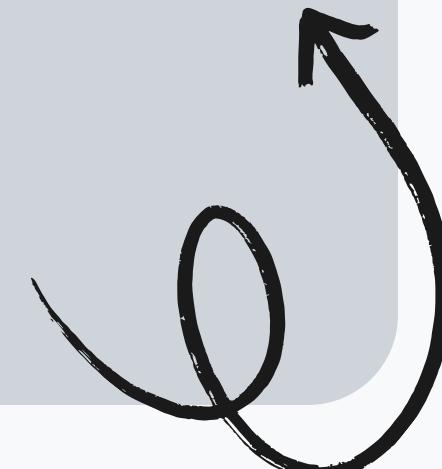
Construcción y análisis:

- El usuario elige una opción.
- El sistema genera una respuesta automática.
- El programa compara resultados.
- Se muestra el ganador.

```
# Función que genera la jugada de la computadora
def obtener_jugada_computadora(opciones):
    # La computadora elige una opción de forma aleatoria
    return random.choice(opciones)

# Función que decide quién gana
def determinar_ganador(usuario, computadora):
    # Si ambos eligen lo mismo, es empate
    if usuario == computadora:
        return "Empate"
    # Condiciones donde el usuario gana
    elif (usuario == "piedra" and computadora == "tijera") or \
        (usuario == "tijera" and computadora == "papel") or \
        (usuario == "papel" and computadora == "piedra"):
        return "¡Ganaste!"
    # En cualquier otro caso, gana la computadora
    else:
        return "Perdiste"

# Función principal que controla el flujo del juego
def iniciar_juego():
    # Lista con las opciones válidas del juego
```



RELACIÓN CON LAS UNIDADES DE LA MATERIA

- Unidad 1: Análisis y diseño del proyecto.
- Unidad 2: Programación y uso de estructuras.
- Unidad 3: Lógica y funciones.
- Unidad 4: Integración y presentación final.



Impacto de la tecnología en la vida

- La tecnología facilita el entretenimiento digital.
- Automatiza procesos similares a decisiones humanas.
- Los videojuegos son una forma común de interacción tecnológica.
- Refleja el avance tecnológico en la sociedad.

Las nuevas tecnologías han transformado nuestra **vida cotidiana**, afectando los métodos de trabajo, comunicación y aprendizaje, facilitando conexiones y acceso a información, aunque también presentan desafíos para la interacción social y la privacidad.



RESULTADOS DEL PROYECTO

- El software funciona correctamente.
- Es fácil de usar y entender.
- Cumple con los objetivos planteados.
- Integra los contenidos de la materia.

```
=====
Juego: Piedra, Papel o Tijera
Simulación de decisiones automatizadas
Impacto de la tecnología en el entretenimiento
=====

Elige piedra, papel o tijera (o escribe 'salir'): piedra
Tú elegiste: piedra
La computadora eligió: piedra
Resultado: Empate
```

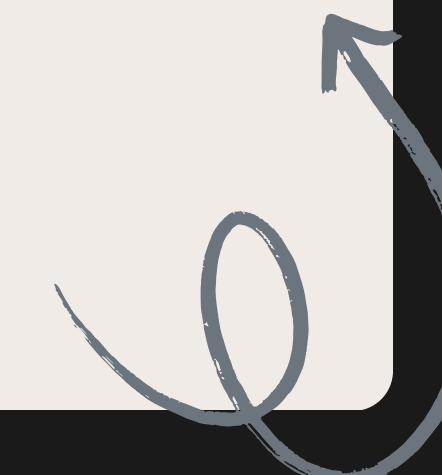


CONCLUSIÓN Y CIERRE

En resumen:

- La tecnología tiene un impacto directo en la sociedad.
- El desarrollo de software ayuda a comprender su funcionamiento.
- Proyectos simples permiten reflexionar sobre el futuro tecnológico.

Agradecimientos



**Gracias por su atención y por ver el
video, espero que haya sido de
vuestra entendimiento y agrado**

Muchas gracias por ver.