

Intro Desarrollo de Videojuegos 2024 - UNQ v.I

2 | Juego y Videojuego - Vectores



¿Por qué queremos hacer videojuegos?

<https://www.menti.com/cwu1praqjt>



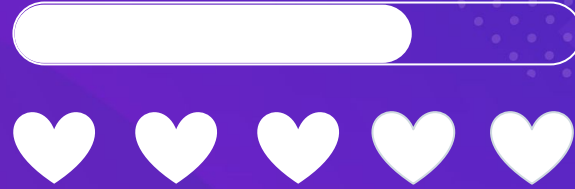
FTL





<http://www.depressionquest.com/>

"The goal of this game is twofold: firstly, we want to illustrate as clearly as possible what depression is like, so that it may be better understood by people without depression. Hopefully this can be something to spread awareness and fight against the social stigma and misunderstandings that depression sufferers face. Secondly, our hope is that in presenting as real a simulation of depression as possible, other sufferers will come to know that they aren't alone, and hopefully derive some measure of comfort from that."



01

Definición de juego

El origen y el desarrollo

Libertad de
elección

01

Límites de
espacio y
tiempo

02

Reglas

03

JUEGO

CONCEPTOS BÁSICOS
DE UN SISTEMA
COMPLEJO

06

EMOCIONES
ESTÉTICA

05

DESAFÍOS
OBJETIVOS

04

Realidad
paralela

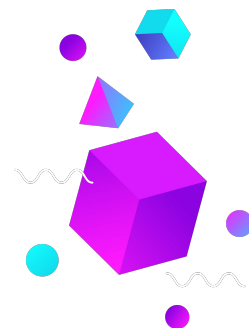
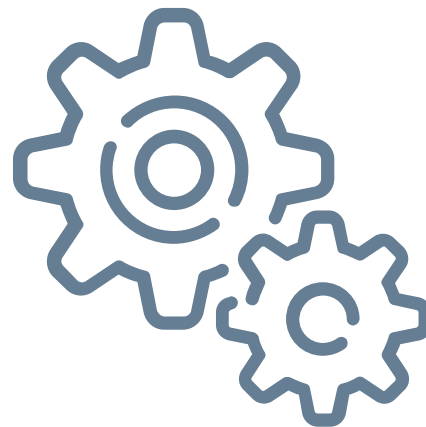
HUIZINGA

- El juego es una acción u ocupación libre, que se desarrolla dentro de unos límites temporales y espaciales determinados, según reglas absolutamente obligatorias, aunque libremente aceptadas, acción que tiene su fin en sí misma y va acompañada de un sentimiento de tensión y alegría y de la conciencia de ser de otro modo que en la vida corriente. Definido de esta suerte, el concepto parece adecuado para comprender todo lo que denominamos juego en los animales, en los niños y en los adultos: juegos de fuerza y habilidad, juegos de cálculo y de azar, exhibiciones y representaciones. Esta categoría, juego, parece que puede ser considerada como uno de los elementos espirituales fundamentales de la vida. (Huizinga, 2007, pp. 45-46)

CAILLOIS

Actividad

- LIBRE
- SEPARADA
- INCIERTA
- IMPRODUCTIVA
- REGLAMENTADA
- FICTICIA

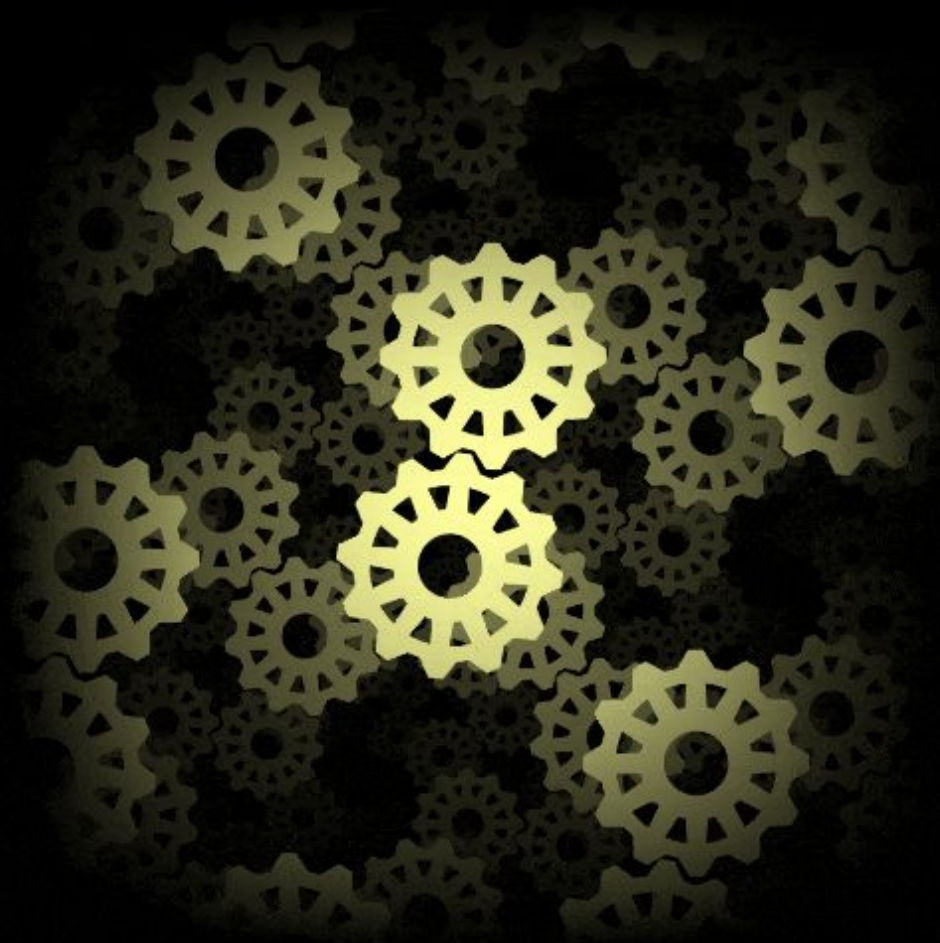


1ER DESAFÍO

Ejercicio de game design

https://www.flippity.net/ra.asp?k=1gXsQPH_CjgfUPEDI-fYEa2k2nw8-T6M1x2iGq1pNy5M





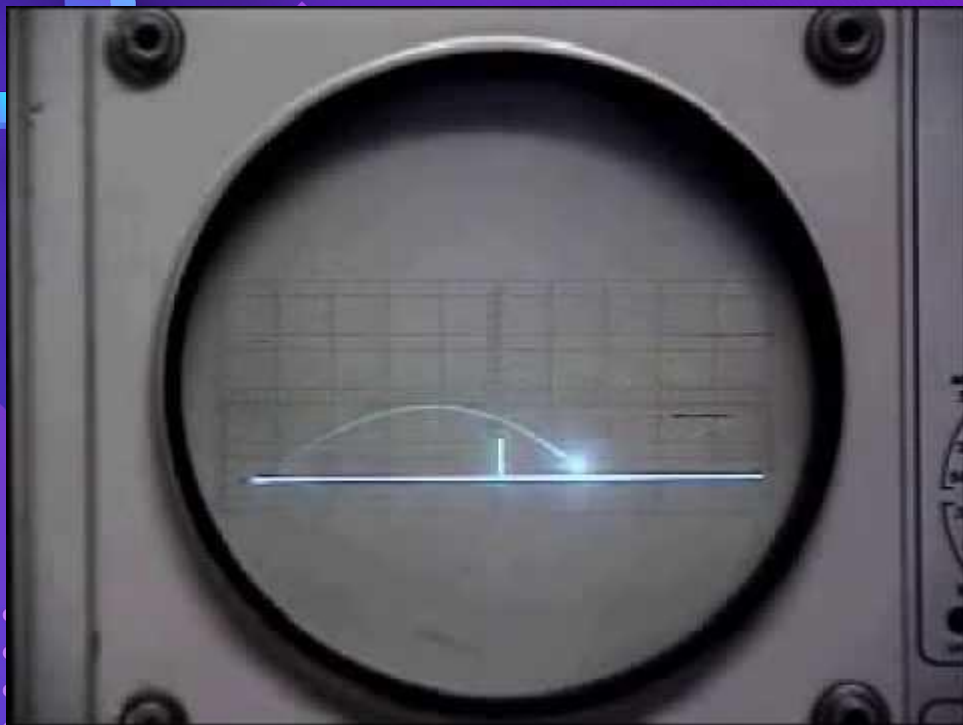
El mundo del juego

HISTORY OF VIDEO GAMES



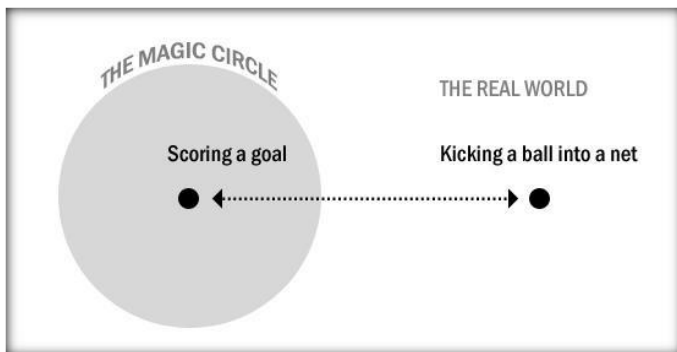
La máquina y el hombre

Manchester Mark I(Universidad de Mánchester, 1948) siendo esta una plataforma para investigar la Inteligencia Artificial ;Spacewar!(Steve Russel, 1962), Bertie the Brain(Josep Kates, 1950), Nimrod(John Makepeace Bennett, 1951) y OXO(Alexander Douglas, 1952)(Djaouti etal., 2011)



La máquina y el hombre

Uno de los primeros videojuegos,
Tennis for two



Huizinga.

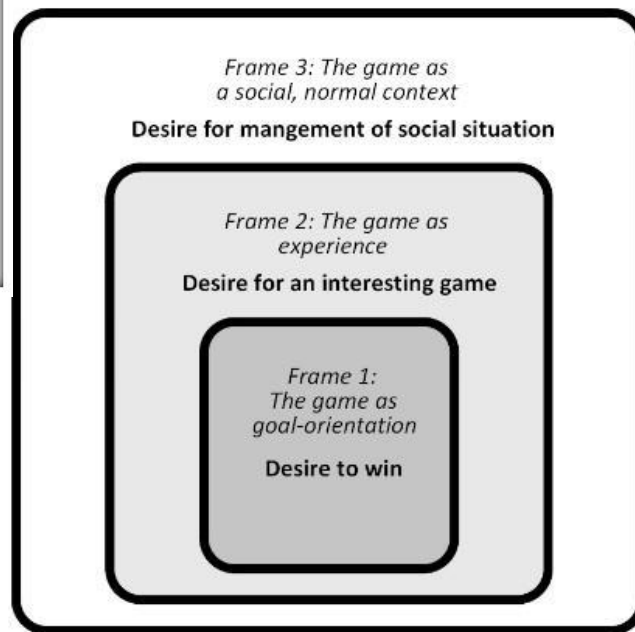


Fig. 2: Three Frames for Every Game Action (Juul 2009)

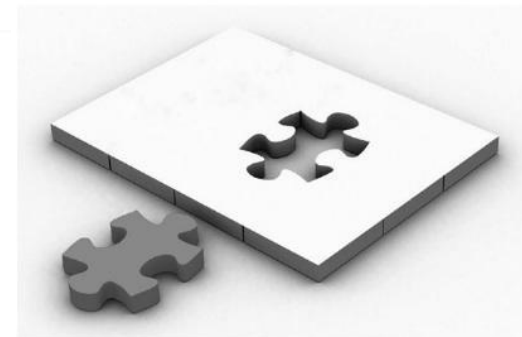


Fig. 3: A Game as a Puzzle Piece that Fits in a Context (Fotolia.com)

Juul

CÍRCULO MÁGICO

Huizinga-Jesper Juul - Coppock







The image is a title screen for a game called "MARBLE LEGENDS". The background is a vibrant, colorful nebula or galaxy with swirling patterns of blue, purple, and pink. Scattered throughout the space are numerous small, bright stars. Several planets or moons are visible: a large, orange-yellow planet with a ring system is positioned in the upper right; a smaller, orange-yellow planet is in the upper center; a small, reddish-brown planet is in the center; and a large, purple planet with a prominent blue diagonal band is in the lower left. The title "MARBLE LEGENDS" is centered in a golden, serif font. The word "MARBLE" is on the left, and "LEGENDS" is on the right, with a small, stylized planet acting as a separator between them. The word "LEGENDS" is underlined with a thin golden line. The entire scene is framed by a black border at the top and bottom.

MARBLE LEGENDS

EL JUEGO



El hombre que juega sigue siendo en el jugar un hombre que se comporta, aunque **la verdadera esencia del juego consista en liberarse** de la tensión que domina el comportamiento cuando se orienta hacia objetivos.

Gadamer

COMPORTAMIENTOS

Representación - autorepresentación

Universo del juego

Realidades distintas

■ Estructura ordenada Reglas

Límites

Jugador

■ Actor

Asume un rol

Medios para la manifestación del juego

Esencia extrapolada

Espectador

Observa las representaciones

■ Medio para la autoevaluación

El juego como medio de entretenimiento para el espectador

el juego

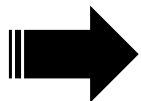
DIMENSIONES DEL JUEGO



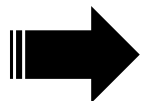


GAMES & RELATED MEDIA

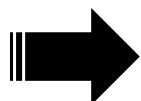
VIDEO GAMES



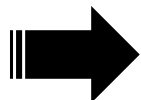
CONSOLE



PC



CLOUD



MOBILE



PUBLISHERS



DIGITAL
DISTRIBUTION



STREAMING
PLATFORMS



ESPORTS



INFLUENCERS /CONTENT CREATORS

ANCILLARY REVENUE



HARDWARE



BOSS

Diversificadores

- Los juegos son para todes
- Ve al punto
- Juegos para todas las edades
- La edad del pixel
- Sonandollega
- Moviendo el objetivo
- Dungeon Master
- Protip
- ¿Este es el fin?
- Juntos a la par
- Feeling it
- La Ley de Murphy
- Aliados improbables
- ¿Crees en la gravedad?
- Con guía en mano
- Un botón por aquí, un lápiz por allá...
- Solo 3
- A toa' velocidá
- Leyenda local
- NOT THE BEES!

Tarea para la clase siguiente (Obligatoria)

Vamos a ejercitar el pensamiento crítico y nuestra capacidad para diseccionar un juego. Para la clase siguiente, se debe:

- Analizar un videojuego a elección y completar la ficha [disponible en este link](#) (*crear una copia personal*).
- Se entrega en un form que les pasaremos por Discord.



Desafío #3

- Generar un proyecto nuevo en GODOT con:
 - Un personaje “Player”, compuesto de un cuadrado negro.
 - 2 inputs de movimiento: “mover_izq” y “mover_der”.
 - Un fondo blanco.

Demo terminada de referencia:

- [Demo ejecutable en HTML](#)

Pueden usar los assets de la demo:

- [Assets demo](#)

O crear los suyos con otro programa:

- [Pixilart](#)



Video tutorial Desafío #3

Intro Desarrollo de Videojuegos 2022 - UNQ v.I

1 | Concepto de juego y videojuego



Desafío #4

- Generar un proyecto en GODOT con:
 - Al Cuadrado negro (player) se le incorpora un nuevo rectángulo amarillo (mano) que apunta al mouse y tira bolitas.
 - Generar un rectángulo naranja como “Torreta” que puede ver al personaje y le dispara.

[Template inicial desafío 4](#)

[Demo objetivo final](#)



Video tutorial Desafío #4



Vectores



Los vectores se utilizan para representar puntos en el espacio, y tienen sobrada importancia sobretodo en física, para representar magnitudes como la **velocidad**, **fuerza**, etc.

• 01

En videojuegos también tenemos infinitos ejemplos en los que se usan estas magnitudes, como puede ser la velocidad de un coche en un juego de carreras o la fuerza que recibe un objeto después de que el jugador choque contra él, etc.

• 02

El uso de vectores en videojuegos proviene del mundo gráfico, donde se los utiliza para representar puntos en la pantalla o para computar la geometría de un objeto 3D y proyectarla.

• 03

Eje de coordenadas

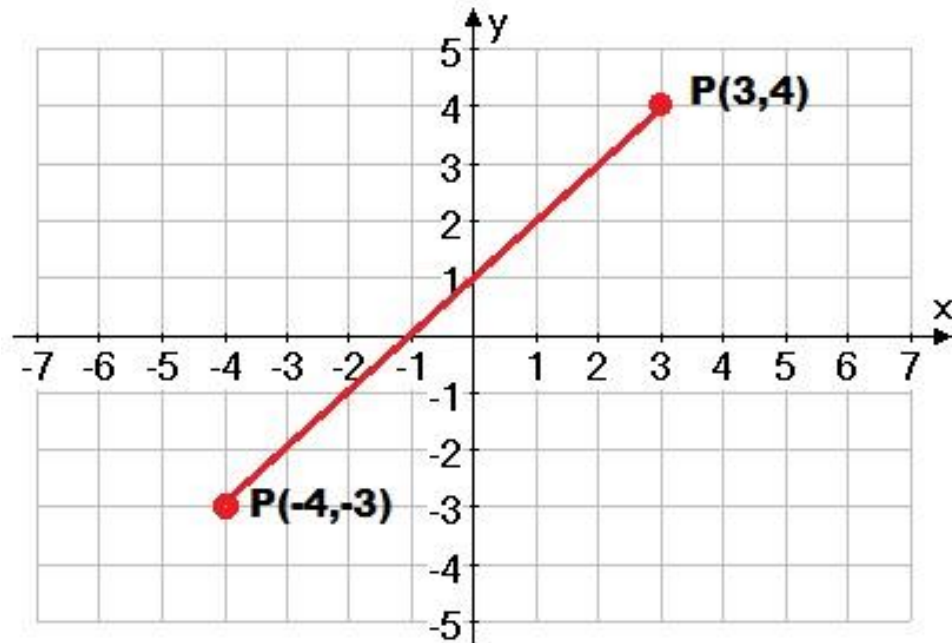


1

Plano cartesiano

Los ejes de coordenadas cartesianas son una parte fundamental de las matemáticas y la física, ya que nos sirven para representar funciones y también posiciones. En ellos, cada eje representa una dimensión, por lo que si hablamos de videojuegos encontraremos dos ejes básicos.

Representado por dos ejes X e Y que indican las dimensiones ancho y alto respectivamente. Se utiliza en juegos en dos dimensiones.

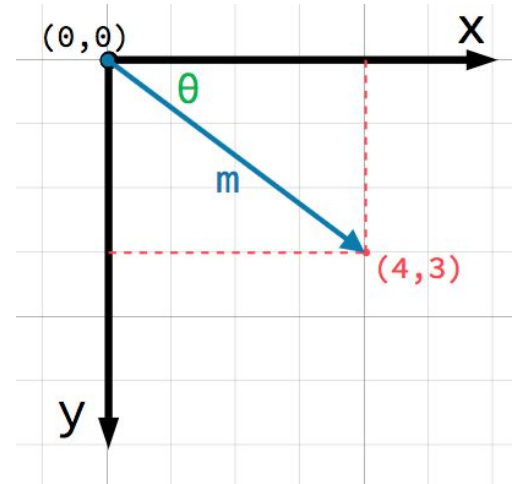
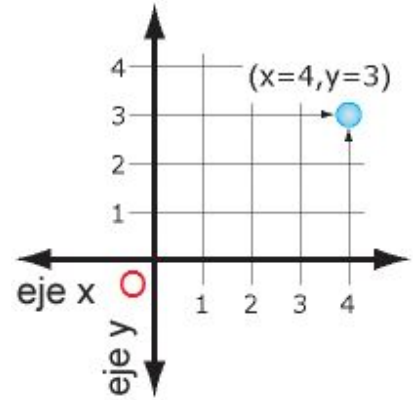
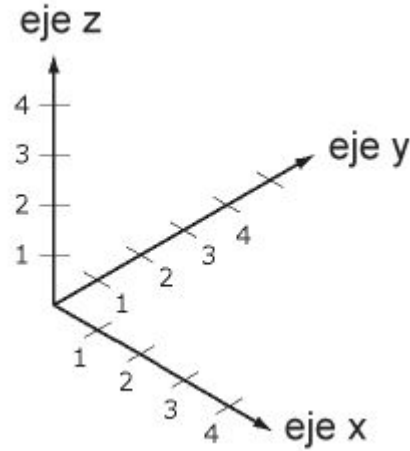




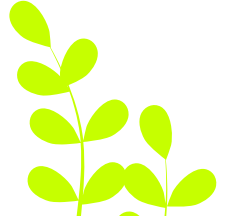
El origen

2

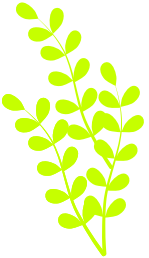
Cualquier posición en el plano 2D puede ser identificada por un par de números de esta manera. Sin embargo, también podemos pensar en la posición $(4, 3)$ como una **compensación** del punto $(0, 0)$, u **origen**. Dibuja una flecha apuntando desde el origen hasta el punto



Tutorial



1. Formato de entrega y versionado



¿Cómo inicio el proyecto?

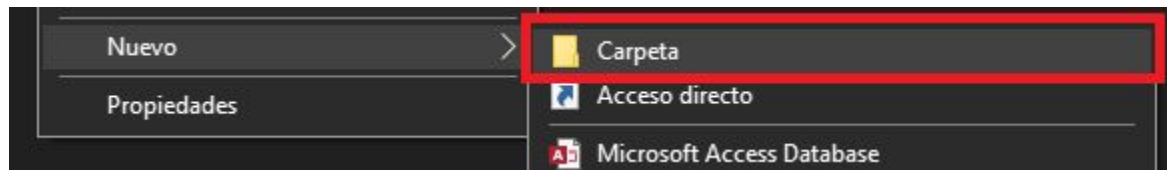
El formato de entrega es:

- El repositorio debe tener nombre “entregas-ipv-apellido-nombre”
- Las entregas deben realizarse en directorios dentro de ese repositorio, siguiendo el formato de nombre de “entrega-x-apellido-nombre”
- El nombre del proyecto de Godot (que se setea en la metadata del proyecto) también debe seguir el formato de las entregas, “entrega-x-apellido-nombre”

¿Cómo lo creamos y versionamos?

En el primer día, vino el repositorio...

El primer paso es crear un directorio raíz que nos servirá como repositorio *local*



Y la renombramos



Luego inicializamos el directorio como un repositorio git (asegurándonos primero de tener [instalado Git](#))

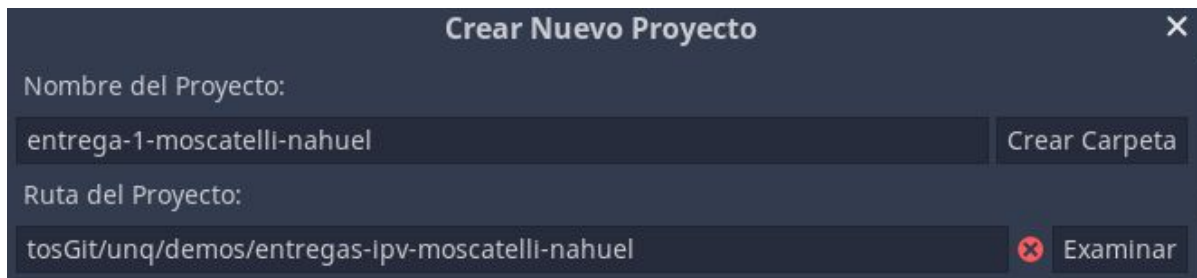
Desde la consola (como Git Bash):

```
R4nKF1v3@DESKTOP-G1DQF91 MINGW64 ~/ProyectosGit/unq/demos/entregas-ipv-moscattelli-nahuel
$ git init
Initialized empty Git repository in D:/ProyectosGit/unq/demos/entregas-ipv-moscattelli-nahuel/.git/
R4nKF1v3@DESKTOP-G1DQF91 MINGW64 ~/ProyectosGit/unq/demos/entregas-ipv-moscattelli-nahuel (main)
$ |
```

¡Ahora, quiero mi proyecto!

Ahora queremos crear el proyecto de Godot. Esto podemos hacerlo directamente desde Godot.

En el launcher de Godot, le damos a 'Nuevo Proyecto' y colocamos el nombre del proyecto acorde al formato



Crear Nuevo Proyecto

Nombre del Proyecto:

entrega-1-moscatelli-nahuel

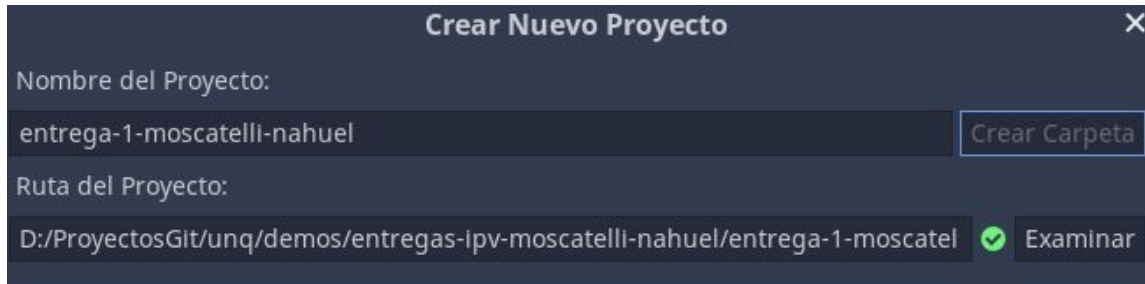
Crear Carpeta

Ruta del Proyecto:

tosGit/unq/demos/entregas-lpv-moscatelli-nahuel

Examinar

Y, asegurandonos que en 'Ruta del Proyecto' esta la ruta del repositorio, le hacemos click a 'Crear Carpeta', lo que nos creará un nuevo directorio en esa ruta con el nombre del proyecto



Crear Nuevo Proyecto

Nombre del Proyecto:

entrega-1-moscatelli-nahuel

Crear Carpeta

Ruta del Proyecto:

D:/ProyectosGit/unq/demos/entregas-lpv-moscatelli-nahuel/entrega-1-moscatel

Examinar

Hecho esto, se puede comenzar a trabajar!

¿Y cómo lo versiono?

Desde la consola chequeamos el status ('git status') en el repositorio local, allí debe de aparecer como 'untracked' el repositorio que corresponde al proyecto de entrega

```
R4nKF1v3@DESKTOP-G1DQF91 MINGW64 ~/ProyectosGit/unq/demos/entregas-ipv-moscatelli-nahuel (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  entrega-1-moscatelli-nahuel/

nothing added to commit but untracked files present (use "git add" to track)
```

Lo agregamos con 'git add <file>' para situarlo en staged

```
R4nKF1v3@DESKTOP-G1DQF91 MINGW64 ~/ProyectosGit/unq/demos/entregas-ipv-moscatelli-nahuel (main)
$ git add entrega-1-moscatelli-nahuel/
```

Y luego hacemos 'git commit -m "mensaje del commit"' para confirmar la transacción

```
R4nKF1v3@DESKTOP-G1DQF91 MINGW64 ~/ProyectosGit/unq/demos/entregas-ipv-moscatelli-nahuel (main)
$ git commit -m "Creada entrega 1"
```

¡Pero yo quiero compartirlo!

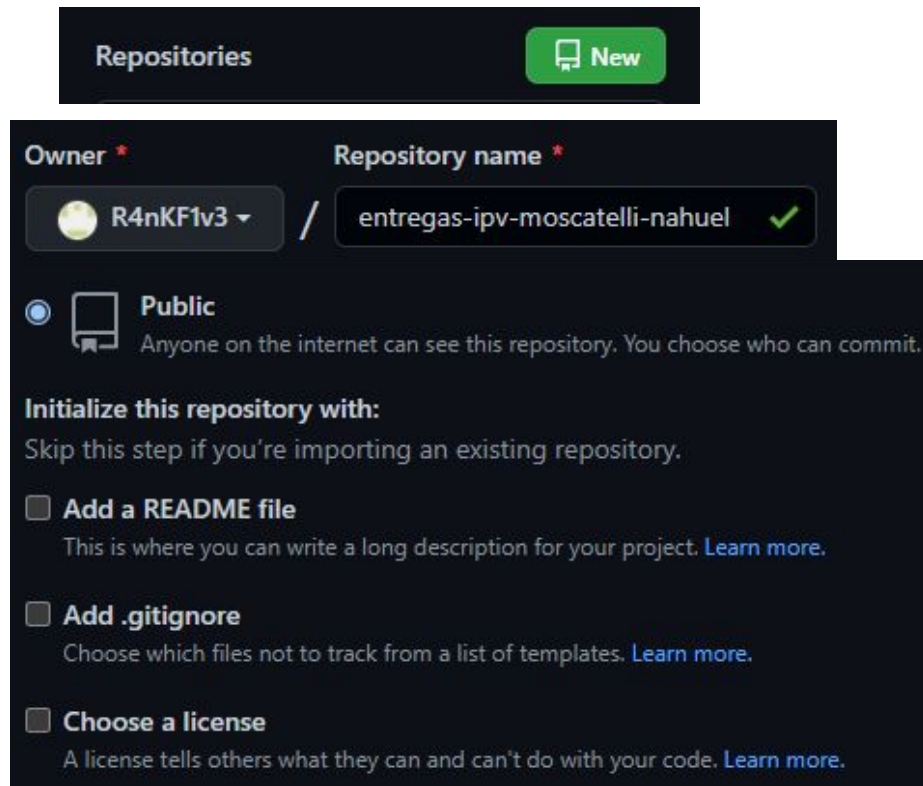
Para eso necesitamos crear un Repositorio Remoto. En la cátedra aconsejamos usar [Github](#), pero alternativas como [Gitlab](#) o similares también son válidas.

Primero buscamos el boton de crear nuevo repositorio en Github (teniendo una cuenta ya creada)

En la pestaña nueva, le ponemos el nombre al repositorio con el formato de entrega

Nos aseguramos que la visibilidad este en “Público”


Y también nos aseguramos que no este tildada ninguna de las opciones referidas a la creación de un README, .gitignore o licencias



The screenshot shows the 'New repository' form on GitHub. At the top, there's a dark header with the word 'Repositories' and a green 'New' button with a repository icon. Below this, the form is divided into sections. The first section has two input fields: 'Owner' with a dropdown menu showing 'R4nKF1v3' and a 'Repository name' field containing 'entregas-ipv-moscatelli-nahuel' with a green checkmark. Below these is a radio button for 'Public' (selected) and a description: 'Anyone on the internet can see this repository. You choose who can commit.' The next section is titled 'Initialize this repository with:' and includes a note: 'Skip this step if you're importing an existing repository.' There are three checkboxes: 'Add a README file' (unchecked), 'Add .gitignore' (unchecked), and 'Choose a license' (unchecked). Each checkbox has a brief description and a 'Learn more' link.

Repositories New

Owner * R4nKF1v3 / Repository name * entregas-ipv-moscatelli-nahuel ✓

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

GITHUB

¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones [Git](#).

¿Para qué sirve?

GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

¡A subir el código!

Ahora tenes un repositorio remoto limpio, pero necesitas vincular tu repositorio local para poder subir tu código

Si seguiste el tutorial hasta ahora, te va a salir una sugerencia automática de github con los comandos a introducir para subir rápido tu repositorio ya existente.

...or push an existing repository from the command line

```
git remote add origin git@github.com:R4nKF1v3/entregas-ipv-moscatelli-nahuel.git
git branch -M main
git push -u origin main
```

Estos comandos funcionan independientemente de la plataforma

¡Y listo! Ahora, cuando quieras subir tu código, solo tenes que repetir el comando 'git push' (sin los demás parámetros) para que lo suba directamente a la branch seleccionada



Gracias!
Nos vemos en la próxima