

Manual de usuario

Este proyecto está desarrollado y pensado para Ubuntu 24.04 y se utilizan varias dependencias que se descargan en el instalador ([ver anexo](#)).

Instalación

Para poder utilizar nuestro software deberás de seguir una serie de pasos específicos.

En primer lugar, bajarse el *.zip* del último release en el [repositorio](#) y descomprimirlo.

Luego podrás ver un archivo bash *install.sh* que funciona de instalador: Cuenta con un pequeño menú que te permitirá ver cuáles dependencias son necesarias e instalarlas todas juntas o por separado, también permite compilar y mover archivos a rutas conocidas (binarios a */bin/*, assets a */var/taller* y configs a */etc/taller*). Como última opción, puedes hacer todo junto: instalar dependencias, compilar y mover archivos.

```
nicolas@nicor:~/TALLER/Counter2D-2025c1-grupo9$ ./install.sh

===== MENÚ DE INSTALACIÓN =====
1. Mostrar información de dependencias
2. Instalar todas las dependencias
3. Actualizar sistema (apt update)
4. Instalar dependencias de audio
5. Instalar dependencias gráficas
6. Instalar dependencias de X11 y Qt6
7. Instalar soporte YAML
8. Compilar proyecto con make
9. Ejecutar tests unitarios
10. Mover binarios a /usr/bin, assets a /var/NAME/ y configs a /etc/NAME/
11. Ejecutar TODO (deps + compilación + tests)
0. Salir
Seleccione una opción: █
```

Una vez terminada la instalación puedes correr los distintos binarios como cualquier otro:

taller_client, *taller_server*, *taller_editor* y *taller_tests*

No hay parámetros para ninguno.

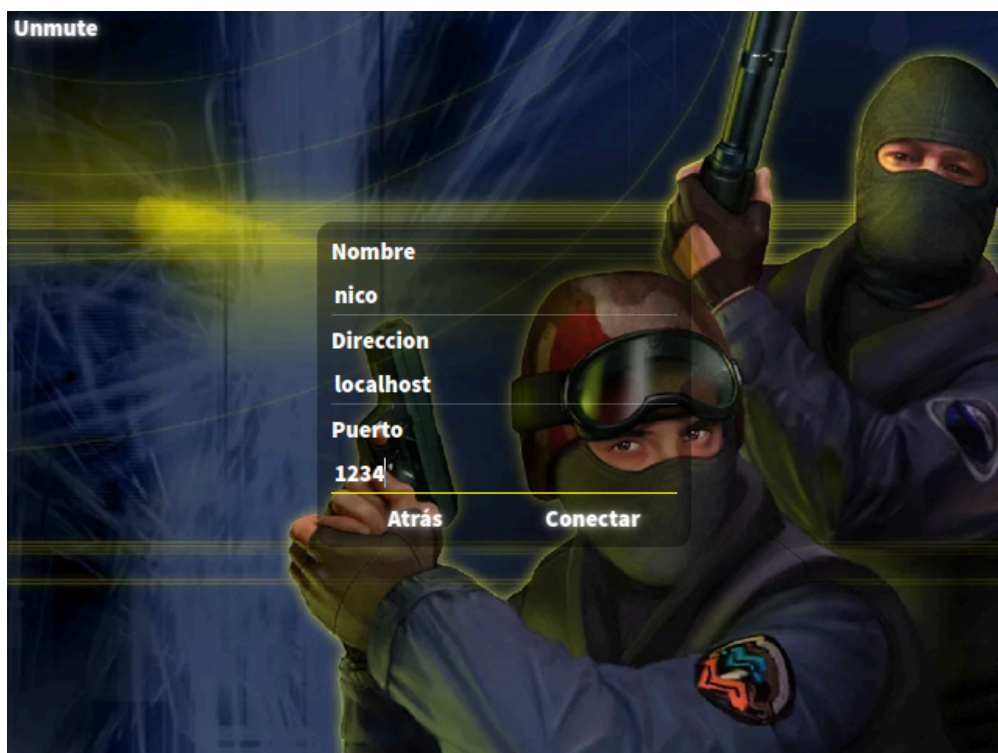
En caso de querer cambiar la configuración que te proveemos puedes cambiar el *config_server.yaml* o *config_client.yaml* según corresponda ([ver anexo](#)).

Servidor

Para levantar el servidor deberás abrir una consola y ejecutar *taller_server*. En caso de querer cerrarlo puedes escribir por consola “**q**” y este cerrará la conexión de todos los clientes, partidas e hilos con los que estaba trabajando.

Cliente

Para ejecutar el cliente deberás abrir una consola y ejecutar *taller_client*. Luego debes poner tu nombre (no puede ser vacío ni alguno que ya esté en uso), dirección y puerto del servidor.

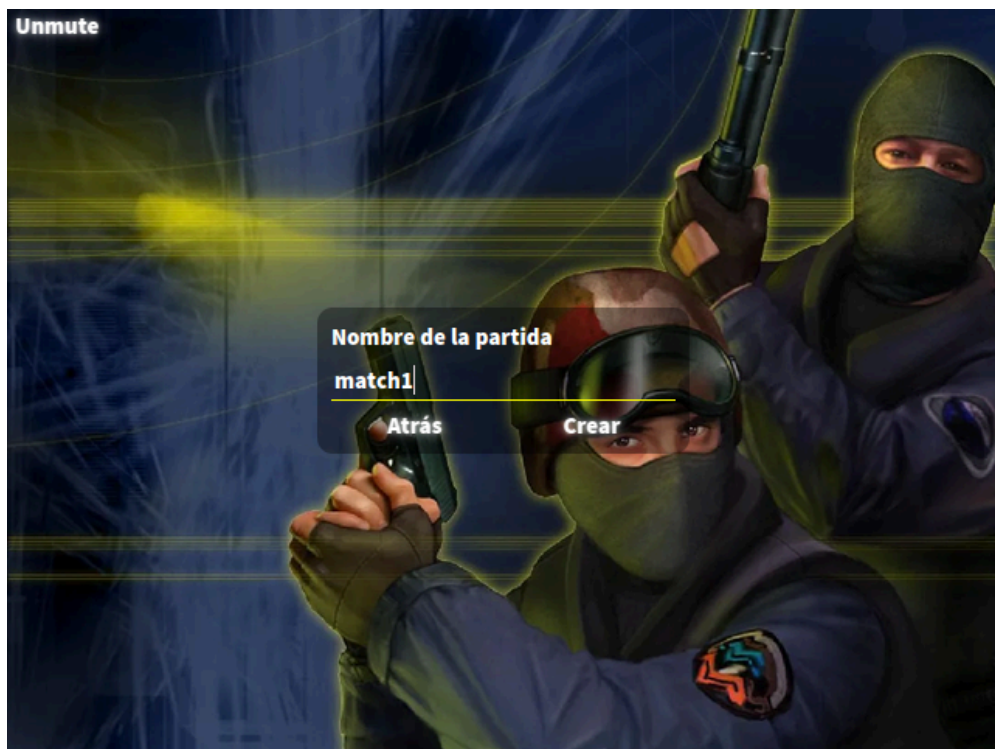


- El nombre no puede ser vacío.
- La dirección es la dirección IP donde vive el host (localhost en este trabajo).
- Puerto es el puerto donde se abrió el servidor.
- El botón de arriba a la izquierda es para silenciar/activar la música del menú.

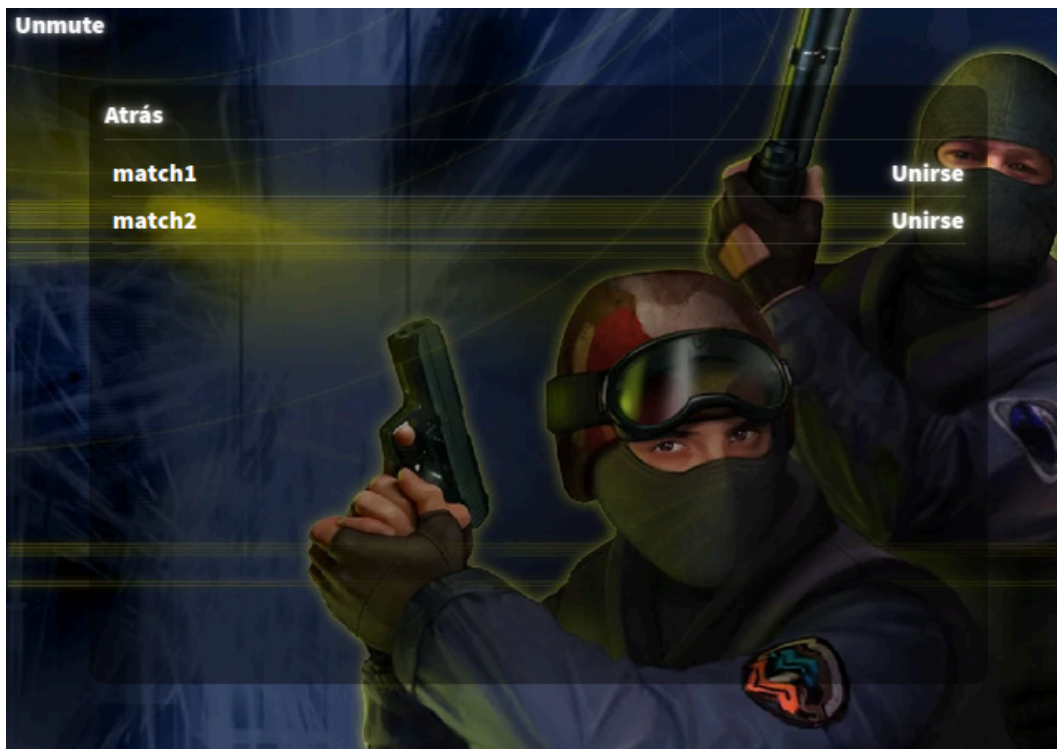


Una vez dentro, podrás elegir entre crear una partida o buscar alguna existente y unirte.

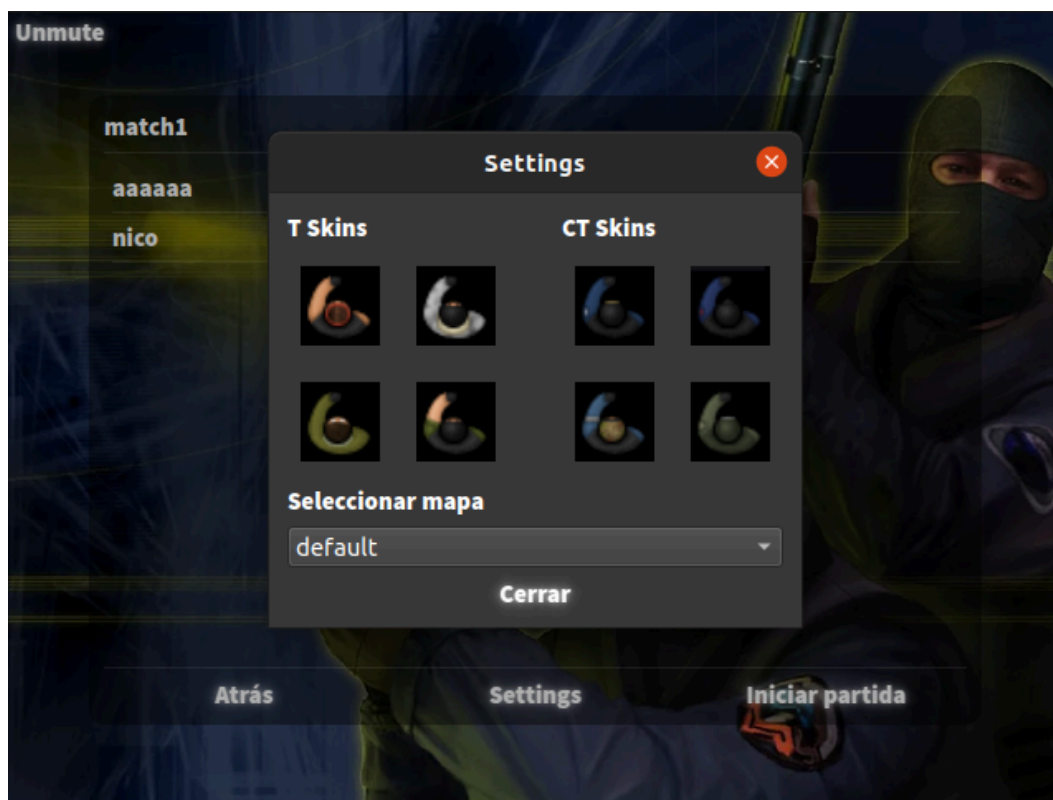
Para crear una partida solo se necesita asignarle un nombre y te mete automáticamente.



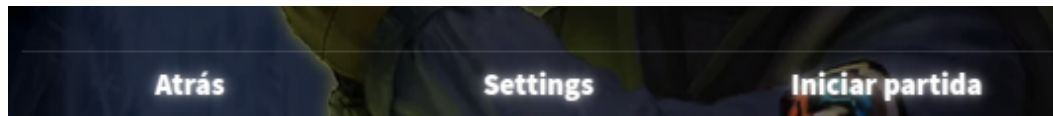
Si lo que quieres es buscar una partida y unirte se te mostrará una lista de todas las partidas que esperan jugadores.



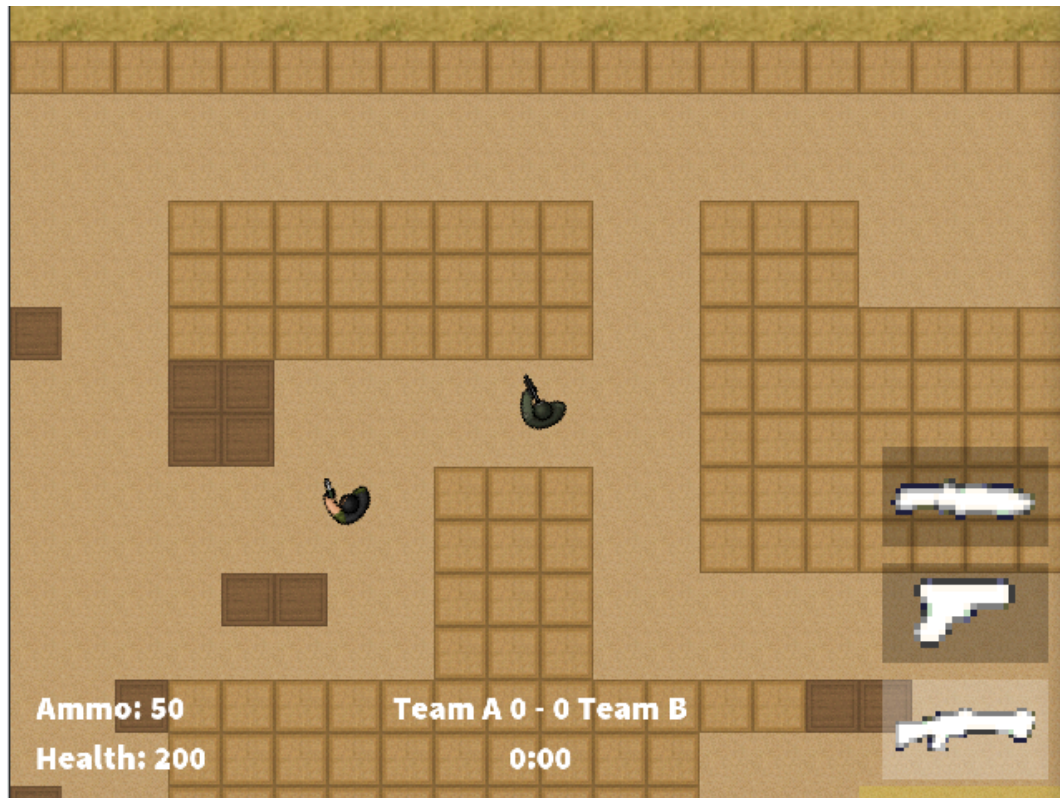
Una vez dentro de una partida en espera tendrás una sección de configuración para elegir skin de terrorista, policía y mapa en el que se quiere jugar.



Luego, cuando la cantidad de jugadores en espera es una cantidad apropiada, se activa el botón de iniciar partida.



Una vez dentro la partida, ya puedes jugar al Counter Strike 2D como dice el enunciado.



Controles de juego:

- **a** *moverse hacia la izquierda*
- **s** *moverse hacia abajo*
- **d** *moverse hacia la derecha*
- **w** *moverse hacia arriba*
- **click izquierdo** *disparar*
- **1** *empuñar arma principal*
- **2** *empuñar arma secundaria*
- **3** *empuñar cuchillo*
- **4** *plantar*
- **e** *defusar*
- **f** *agarrar algo del piso*
- **mover puntero** *apuntar*

- **m** *mutear/desmutear el sonido*
- **b** *abrir/cerrar tienda*
- **g** *habilitar/deshabilitar limitación del campo de visión*

Editor

Al momento de la creación de este documento no estaba listo el editor para sacar capturas

Para ejecutar el editor deberás abrir una consola y ejecutar *taller_editor*. Una vez iniciado, podrás elegir entre crear un mapa desde cero o editar uno ya existente. Al crear un mapa, se te pedirá que indiques un nombre y la cantidad de personas para las que está pensado. Si elegís editar un mapa, el proceso se divide en dos fases: una para colocar los puntos de aparición (spawns) y las zonas de bomba, y otra para colocar bloques. El editor permite desplazarse utilizando la rueda del ratón, usar el clic izquierdo para colocar bloques y el clic derecho para borrarlos. También contarás con botones para guardar el mapa y para agregar filas o columnas, siendo el tamaño inicial de 30x30.

ANEXO: Dependencias

El proyecto requiere instalar varias bibliotecas de desarrollo para compilar correctamente. Este listado refleja lo que instala el script *install.sh* más las dependencias adicionales necesarias que se descargan automáticamente al compilar.

Dependencias de audio

Requeridas para reproducción y procesamiento de audio (usadas en fluidsynth, libxmp, opus, etc.):

- libopus-dev
- libopusfile-dev
- libxmp-dev
- libfluidsynth-dev
- fluidsynth
- libwavpack1
- libwavpack-dev
- wavpack

Dependencias gráficas y tipográficas

Para renderizado de texto y gráficos vectoriales:

- libfreetype-dev
- libfreetype6-dev
- libfontconfig1-dev

Dependencias de X11 y Qt6

Requeridas para interfaces gráficas y soporte de ventanas:

- libx11-dev
- libx11-xcb-dev
- libxext-dev
- libxfixes-dev
- libxi-dev
- libxrender-dev
- libxcb1-dev
- libxcb-glx0-dev
- libxcb-keysyms1-dev
- libxcb-image0-dev
- libxcb-shm0-dev
- libxcb-icccm4-dev
- libxcb-sync-dev
- libxcb-xfixes0-dev
- libxcb-shape0-dev
- libxcb-randr0-dev
- libxcb-render-util0-dev
- libxkbcommon-dev
- libxkbcommon-x11-dev
- qt6-base-dev
- qt6-multimedia-dev

Multimedia (GStreamer)

Requerido por Qt Multimedia para reproducir audio/video:

- gstreamer1.0-plugins-base
- gstreamer1.0-plugins-good
- gstreamer1.0-plugins-bad
- gstreamer1.0-plugins-ugly
- gstreamer1.0-libav
- gstreamer1.0-tools

YAML

Requerido por el servidor, cliente, editor y tests para cargar configuraciones:

- libyaml-cpp-dev

SDL2 y SDL2pp

El cliente utiliza SDL2 y libSDL2pp para manejar gráficos 2D, entrada del usuario, audio y texto de forma eficiente y moderna:

- SDL2
- SDL2_image
- SDL2_mixer
- SDL2_ttf
- libSDL2pp

Google Test (solo para tests)

Se descarga automáticamente al compilar con -DTALLER_TESTS=ON, no requiere instalación manual.

Herramientas de compilación

Estas herramientas no son parte del script install.sh, pero son necesarias para compilar correctamente:

- cmake >= 3.24
- make
- g++ con soporte para C++20
- git
- pkg-config

ANEXO: Configuración

Config_Server

server:

port: "1234" string de números con el puerto
tick_rate: 60 ticks por segundo que maneja el game loop
max_events_per_tick: 32 cantidad máxima de eventos que se procesan en un loop

game:

max_players_per_team: 5 cantidad máxima permitida de jugadores por equipo
min_players_per_team: 0 cantidad mínima permitida de jugadores por equipo
rounds_until_role_change: 1 cantidad de rondas hasta invertir roles (atacante y defensor)
rounds_until_end_game: 2 duración en rondas del juego
max_health: 200 vida máxima del jugador
max_bullets: ojo al piojo
speed: 4.0 velocidad del jugador
purchase_duration: 10.0 duración de la fase de compra
bomb_duration: 20.0 duración de la bomba una vez plantada hasta que explota
time_to_plant: 30.0 duración de la fase de plantado
time_until_plant: 3.0 tiempo que debe mantener apretado para plantar bomba
time_until_defuse: 3.0 tiempo que debe mantener apretado para defusear bomba
time_until_new_round: 10.0 tiempo de fase post ronda, antes de comenzar una nueva
time_until_end_running: 10.0 tiempo al finalizar el juego antes de que se cierre
ammo_price: 40 precio del cargador de cualquier arma, mas alla de si tiene o no alguna bala
money_winner: 150 dinero que gana cada jugador del equipo ganador al finalizar ronda
money_loser: 100 dinero que gana cada jugador del equipo perdedor al finalizar ronda
initial_money: 1000 dinero inicial de cada jugador
initial_primary_ammo: 0 balas iniciales de arma primaria
initial_secondary_ammo: 0 balas iniciales de arma secundaria

weapons:

weapon_name:
price: 100 precio del arma
min_damage: 15 daño mínimo del arma (sin tener en cuenta distancia)
max_damage: 25 daño máximo del arma (sin tener en cuenta distancia)
burst_fire: true booleano de si el arma dispara o no en rafaga
bullets_per_burst: 3 cantidad de balas por rafaga
burst_delay: 0.05 tiempo entre balas en una rafaga
spread_angle: 0.0 apertura de las balas en un disparo, si es 0 respeta el ángulo
bullets_per_shoot: 1 balas por disparo, la escopeta es la unica que tiene mas de 1
max_range: 40.0 máxima distancia que alcanza una bala si no impacta con nada
max_ammo: 30 cantidad de balas máxima que tiene un arma
cooldown: 0.4 tiempo mínimo que debe pasar entre disparos

Config_Client

client:

fullscreen: false booleano para iniciar el juego en pantalla completa

resolution:

width: 640 ancho de la pantalla al abrir

height: 400 alto de la pantalla al abrir

fov:

angle_deg: 90 ángulo del campo de visión

opacity: 0.75 opacidad del campo nublado