

Guía de Instalación, Compilación y Manual del Juego



Manuel Pato Ibáñez	110640
Lucas Nicolás Pagani	110777
Federico Zanor	112097
Joaquín Schapira	112459

3 de Diciembre de 2025

Índice

1. Instalación, Compilación y Ejecución	3
1.1. Requisitos del Sistema	3
1.2. Clonar el repositorio	3
1.3. Otorgar permisos de ejecución al instalador	3
1.4. Instalar el proyecto	3
1.5. Descargar dependencias	4
1.6. Compilar manualmente	4
1.7. Ejecución de tests manualmente	4
1.8. Ejecución del proyecto	4

1.9. Ejecución con Valgrind (opcional)	5
2. Manual del Juego	6
2.1. Editor de Mapas	6
2.1.1. Colocación de elementos	6
2.1.2. Guardado de mapas	7
3. Iniciar el Juego	7
3.1. Servidor	7
3.2. Cliente	8
3.3. Crear Partida	8
3.4. Unirse a Partida	9
3.5. Pre-partida / Lobby	9
4. Controles del Juego	10
4.1. Movimiento	10
4.2. Sonido	10
4.3. Cheats	10
5. Dinámica del Juego	10
5.1. Objetivo	10
5.2. Sistema de vida	10
5.3. Condición de victoria	11
6. Menú de Mejoras	11
7. Sistema de Ranking	11
8. Página web	12
9. Video Ejemplo	12

1. Instalación, Compilación y Ejecución

Antes de utilizar el juego, es necesario instalar las dependencias y compilar el proyecto. Todo el proceso se encuentra automatizado mediante el script `needforspeed2d.sh`.

1.1. Requisitos del Sistema

- Sistema operativo compatible:
 - Ubuntu 24.04 o distribuciones Linux derivadas.

1.2. Clonar el repositorio

```
git clone https://github.com/Manupato/Need4Speed2D
```

1.3. Otorgar permisos de ejecución al instalador

```
chmod +x needforspeed2d.sh
```

1.4. Instalar el proyecto

Este paso debe ejecutarse como root porque instala los binarios y copia assets/config:

```
sudo ./needforspeed2d.sh install
```

- ¿Qué hace internamente?
 - Instala dependencias esenciales
 - Compila el proyecto
 - Ejecuta los tests del protocolo
 - Instala los ejecutables en el sistema:
 - /usr/bin/taller_client
 - /usr/bin/taller_server
 - /usr/bin/taller_editor
 - Copia recursos del juego a /var/need4speed
 - Copia la configuración a /etc/need4speed
 - En cambio, los mapas jugables se guardan en:
~/.local/share/need4speed/mapas_jugables

En los archivos de configuración, se pueden modificar parámetros como duración de la carrera, características de los autos y tiempos de pantallas intermedias. No es necesario modificar nada para jugar.

1.5. Descargar dependencias

Este comando solo es necesario si desarrollás cosas nuevas, y sumás nuevas librerías que necesiten ser descargadas y no se deseé hacer el install completo. Tener en cuenta, que los cambios solo se verán aplicados al ejecutar el programa desde la raíz y no con el ejecutable creado por el install (Ver inciso 1.8 para entender todas las opciones de ejecución). De estar satisfecho con los cambios y querer aplicarlos al ejecutable, se debe hacer el install nuevamente

Instalar todas las librerías necesarias (SDL, Qt, CMake, entre otras):

```
sudo ./needforspeed2d.sh download
```

1.6. Compilar manualmente

Al igual que el inciso 1.5 esta compilación solo es necesaria si desarrollás cosas nuevas o limpiás build. Sino, con la instalación del inciso 1.3 ya es suficiente. Los cambios aplicados con el build, solo se verán aplicados si se ejecuta desde la raíz, de querer aplicarlos al ejecutable, se debe hacer el install nuevamente.

```
./needforspeed2d.sh build
```

Compilación limpia:

```
./needforspeed2d.sh build --clean  
./needforspeed2d.sh build -c
```

1.7. Ejecución de tests manualmente

Los tests se deben ejecutar desde la raíz del repositorio:

```
./needforspeed2d.sh tests
```

1.8. Ejecución del proyecto

El sistema incluye tres ejecutables:

- Cliente
- Servidor
- Editor de mapas

Si se instalaron (Inciso 1.4), se pueden ejecutar mediante los siguientes comandos desde cualquier terminal:

Editor:

```
taller_editor
```

Servidor:

```
taller_server 8080
```

Cliente:

```
taller_client localhost 8080
```

Tambien, si no se hizo la instalación pero sí la descarga de dependencias (Inciso 1.5) y compilación manual (Inciso 1.6) se pueden ejecutar de forma local desde la raíz del repositorio con los siguientes comandos (Como se entiende que estos comandos son para pruebas de desarrolladores, se harcodea el host y el puerto en localhost y en 8080 respectivamente.):

Cliente:

```
./needforspeed2d.sh run client
```

Servidor:

```
./needforspeed2d.sh run server
```

Editor:

```
./needforspeed2d.sh run editor
```

1.9. Ejecución con Valgrind (opcional)

El cliente y el editor utilizan el directorio **Valgrind_helpers** para supresiones. Igualmente, existen algunos errores de las librerías utilizadas (QT y SDL) que no se pudieron suprimir, pero son errores propios de las mismas.

El servidor no requiere supresiones.

Para usar estos comandos de ejecución con Valgrind, se deben utilizar desde la raíz del repositorio:

Cliente:

```
./needforspeed2d.sh run client --valgrind
```

Servidor:

```
./needforspeed2d.sh run server --valgrind
```

Editor:

```
./needforspeed2d.sh run editor --valgrind
```

A estos comandos, se le puede agregar como parametro --clear para limpiar las supresiones actuales y forzar a generarlas nuevamente. Por ejemplo:

Cliente:

```
./needforspeed2d.sh run client --valgrind --clear
```

2. Manual del Juego

El juego incluye mapas jugables precargados para comenzar a jugar sin configuración adicional. Además, el usuario puede diseñar mapas personalizados utilizando el editor interactivo. Si es su primera vez, se recomienda explorar el editor para crear o modificar un mapa a gusto.

2.1. Editor de Mapas

Para ejecutar el editor:

```
taller_editor
```

El usuario puede elegir entre tres ciudades base:

- **San Andreas:** avenidas amplias y rectas largas.
- **Vice City:** calles angostas y trazados cortos.
- **Liberty City:** un río central, una isla y múltiples puentes.

2.1.1. Colocación de elementos

En la barra superior pueden colocarse:

- Salida
- Meta
- Checkpoints

El cursor indica si la ubicación es válida. La salida ocupa un área considerable; la meta y los checkpoints son más flexibles. Los elementos pueden rotarse con la tecla **R**.

Para corregir errores se dispone de:

- Herramienta **Borrar**
- Deshacer (**Ctrl + Z**)



Figura 1: Vista del editor de mapas.

2.1.2. Guardado de mapas

Opciones disponibles:

- **Guardar como:** crear un nuevo mapa.
- **Abrir mapa jugable:** modificar uno existente.
- **Guardar:** actualizar el mapa actual.

3. Iniciar el Juego

3.1. Servidor

Para iniciar el servidor:

```
taller_server 8080
```

Para finalizar el servidor basta con ingresar la letra q por consola

3.2. Cliente

Cada jugador debe ejecutar:

```
taller_client localhost 8080
```

Se abrirá una ventana con las opciones:

- Crear partida
- Unirse a partida

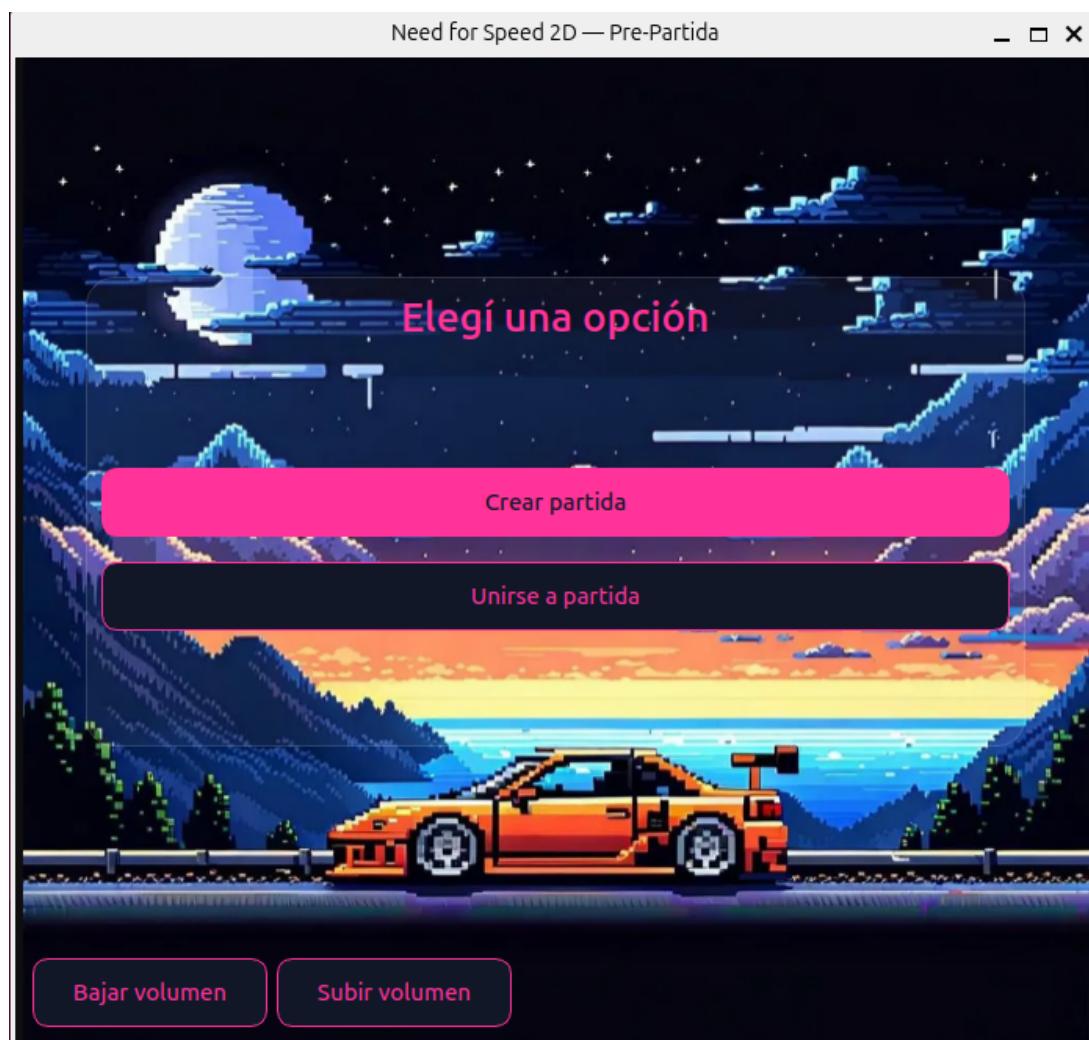


Figura 2: Vista del pre lobby.

3.3. Crear Partida

El anfitrión podrá:

- Seleccionar nombre

- Elegir auto
- Seleccionar mapas jugables

Al confirmar se genera un código de 4 dígitos para que otros jugadores se unan.

3.4. Unirse a Partida

El jugador deberá ingresar:

1. Su nombre
2. Su auto
3. El código de partida

3.5. Pre-partida / Lobby

En la sala de espera se mostrarán los jugadores y sus autos. La partida comenzará cuando algún jugador le de al botón de iniciar partida.



Figura 3: Vista del lobby.

4. Controles del Juego

4.1. Movimiento

W Avanzar
A Girar a la izquierda
S Retroceder / Marcha atrás
D Girar a la derecha

4.2. Sonido

M Silenciar sonidos
N Subir volumen
B Bajar volumen

4.3. Cheats

P Forzar derrota
O Forzar victoria
I Vida infinita
U Modo fantasma

5. Dinámica del Juego

5.1. Objetivo

1. Pasar por todos los checkpoints.
2. Cruzar la meta.
3. Evitar pasto y veredas para no perder velocidad.

5.2. Sistema de vida

El auto posee puntos de vida. Los choques reducen estos puntos según la gravedad del impacto. Si el auto llega a 0:

- Se detiene.
- Explota.
- Recibe una penalización de tiempo.

5.3. Condición de victoria

Gana quien tenga el menor tiempo acumulado al finalizar todos los circuitos.

6. Menú de Mejoras

Entre carreras, los jugadores pueden mejorar su auto. Cada mejora conlleva una penalización de tiempo, por lo que debe elegirse estratégicamente.

7. Sistema de Ranking

El ranking final se calcula sumando los tiempos totales de cada jugador.

Ejemplo

Carrera	Jugador A	Jugador B
1	10s	11s
2	15s	16s
3	25s	20s

Resultado final: Jugador A = 50s

Jugador B = 47s (ganador)



Figura 4: Tabla de posiciones.

8. Página web

El proyecto cuenta con su página web oficial donde se podrá observar un gameplay y tutoriales tanto de instalación como de jugabilidad:

https://joaquinschapira.github.io/taller_TP_pagina_web/

9. Video Ejemplo

Se incluye un video demostrativo del funcionamiento completo del sistema y una partida completa:

<https://www.youtube.com/watch?v=uabR9KEANGU>