

# Puertas y munición en DOOM

*Reglas funcionales observables del subsistema, con criterios de aceptación*

**Sobre este documento.** Esta especificación sigue la estructura de una **SRS ISO/IEC/IEEE 29148** (el estándar formal de requisitos de software vigente, que sustituye a IEEE 830), con los criterios de aceptación redactados en **Gherkin** (Dado / Cuando / Entonces), que es como hoy se automatizan las pruebas. Cada regla cuantitativa conserva su referencia `archivo:línea` al código fuente del que se derivó.

## Índice

|                                                                      |   |
|----------------------------------------------------------------------|---|
| 1. Introducción .....                                                | 2 |
| 1.1 Propósito .....                                                  | 2 |
| 1.2 Alcance .....                                                    | 2 |
| 1.3 Definiciones .....                                               | 2 |
| 1.4 Referencias .....                                                | 2 |
| 2. Requisitos funcionales .....                                      | 2 |
| R1. Una puerta normal se abre, espera y se cierra sola .....         | 2 |
| R2. La puerta espera 3,00 segundos arriba antes de cerrarse .....    | 2 |
| R3. La puerta normal se mueve a 3 unidades de mapa por tic .....     | 3 |
| R4. Las puertas rápidas van cuatro veces más rápido .....            | 3 |
| R5. Una puerta con cerradura exige la llave del color correcto ..... | 3 |
| R6. Una puerta que baja y te atrapa se reabre .....                  | 3 |
| R7. Un enemigo puede abrir puertas normales pero no cerrarlas .....  | 3 |
| R8. Hay puertas que se cierran o abren solas por tiempo .....        | 3 |
| R9. Una puerta se puede abrir de tres maneras .....                  | 3 |
| R10. Una bala de arma de bala hace 7, 14 o 21 de daño .....          | 4 |
| R11. Cada disparo gasta una bala .....                               | 4 |
| R12. Un cargador da 15 balas .....                                   | 4 |
| 3. Tabla de parámetros de negocio .....                              | 4 |
| 4. Historias de usuario y criterios de aceptación .....              | 5 |
| HU-1. Cruzar una puerta normal .....                                 | 5 |
| HU-2. Puerta con llave .....                                         | 5 |
| HU-3. Puerta rápida .....                                            | 5 |
| HU-4. Disparar con un arma de bala .....                             | 5 |
| 5. Incertidumbre declarada .....                                     | 6 |

# 1. Introducción

## 1.1 Propósito

Este documento especifica el comportamiento funcional **observable** de un subsistema de DOOM: la apertura y cierre de puertas de un nivel y el daño y la munición de las armas de bala. Describe reglas de negocio y criterios de aceptación, no diseño interno ni código. La audiencia es analista funcional, QA y responsable de producto.

## 1.2 Alcance

El alcance es lo que un jugador puede observar sin ver el código: qué hace una puerta al usarla, cuánto espera, qué exige una puerta con llave, de cuántas formas se abre, cuánto daño hace una bala y cuánta munición da un cargador. Queda **fuera de alcance** la geometría concreta de cada nivel (que depende del WAD cargado) y todo detalle de implementación interna.

## 1.3 Definiciones

**Tic** paso de simulación del motor. DOOM corre a  $TICRATE = 35$  tics por segundo `doomdef.h:122` las conversiones a segundos de este documento dividen por 35.

**Unidad de mapa** unidad de distancia del motor con la que se expresan velocidades.

**Puerta rápida (blazing)** variante de puerta que va a cuádruple velocidad frente a la normal.

**Special** número que marca una línea del nivel como puerta, puerta con llave, disparador, etc.

**Build pública** la versión de DOOM que documenta la Doom Wiki. Se cita solo **como contraste**, para decir qué NO hay que esperar de este árbol, nunca como la verdad de este código.

## 1.4 Referencias

Especificación derivada leyendo `codigo-mutado/linuxdoom-1.10/`, commit de corte `a77dfb96cb91780ca334d0d4cfd86957558007e0`. El documento hermano de arquitectura (**Documentación técnica: el ciclo de una puerta**, formato arc42) cubre el diseño interno.

# 2. Requisitos funcionales

Cada requisito es una regla verificable, con su evidencia en el código, su nivel de confianza y la forma de comprobarla observando el sistema en ejecución.

## R1. Una puerta normal se abre, espera y se cierra sola

El sistema deberá, al usar una puerta normal, subirla, mantenerla abierta un tiempo fijo y bajarla sola. confianza: alta

Evidencia: ciclo del animador de puertas, estados  $1 \rightarrow 0 \rightarrow -1$  `p_doors.c:168,69,118`.

Comprobación: abrir una puerta y observar que se cierra sin volver a tocarla.

## R2. La puerta espera 3,00 segundos arriba antes de cerrarse

El sistema deberá mantener la puerta normal abierta **105 tics** antes de iniciar el cierre.

confianza: alta

Evidencia: `VD00RWAIT = 105 tics` `p_spec.h:365`, cargado en `door->topwait` `p_doors.c:466` y usado como cuenta atrás al llegar arriba `p_doors.c:182`.  $105 \div 35 = 3,00$  s.

Comprobación: cronometrar desde que la puerta queda del todo abierta hasta que arranca a bajar; deben ser 3,0 s (no 4,3 s, la espera de la build pública).

### R3. La puerta normal se mueve a 3 unidades de mapa por tic

El sistema deberá mover la puerta normal a 3 unidades de mapa por tic. confianza: alta

Evidencia:  $VD00RSPEED = \text{FRACUNIT} * 3$  p\_spec.h:364, asignada a door->speed p\_doors.c:465 y consumida por el motor de movimiento p\_doors.c:171 con FRACUNIT igual a una unidad de mapa m\_fixed.h:35-36.

Comprobación: medir los tics de subida sobre una distancia conocida en un nivel de prueba.

### R4. Las puertas rápidas van cuatro veces más rápido

El sistema deberá mover las puertas rápidas (blazing) a cuádruple velocidad respecto a la normal, es decir 12 unidades por tic. confianza: alta

Evidencia:  $speed = VD00RSPEED * 4$  p\_doors.c:487,492 (y p\_doors.c:300,325 por la vía remota).

Comprobación: comparar el tiempo de apertura de una puerta rápida (special 117/118) con una normal en el mismo hueco.

### R5. Una puerta con cerradura exige la llave del color correcto

El sistema deberá impedir la apertura de una puerta con llave si el jugador no lleva la tarjeta o calavera del color correspondiente, y deberá emitir un sonido de negativa (sfx\_oof).

confianza: alta

Evidencia: azul p\_doors.c:376-381, amarilla p\_doors.c:389-395, roja p\_doors.c:403-408 los tres colores admiten tarjeta o calavera p\_doors.c:376.

Comprobación: intentar abrir una puerta con llave sin tenerla; no abre y suena el gruñido.

### R6. Una puerta que baja y te atrapa se reabre

El sistema deberá, si una puerta que baja atrapa a algo debajo, invertir el movimiento y volver a subir sin causar daño, salvo en las puertas de tipo close y blazeClose. confianza: alta

Evidencia: rebote p\_doors.c:151-164 el motor pasa el movimiento sin daño y devuelve «obstruido» p\_floor.c:156-162.

Comprobación: quedarse bajo una puerta que se cierra y ver que vuelve a subir sin restar vida.

### R7. Un enemigo puede abrir puertas normales pero no cerrarlas

El sistema deberá, en una puerta ya activa, no forzar el cierre cuando quien la usa no es el jugador. confianza: alta

Evidencia: p\_doors.c:430-431.

Comprobación: observar a un enemigo cruzar una puerta; no la cierra sobre sí mismo.

### R8. Hay puertas que se cierran o abren solas por tiempo

El sistema podrá programar puertas sin intervención del jugador: una que se cierra a los 30 segundos y otra que se abre a los 5 minutos; además, el tipo close30ThenOpen se cierra, espera 30 segundos abajo y reabre. confianza: media

Evidencia: cierre a 30 s p\_doors.c:521, apertura a 5 min p\_doors.c:549, espera de reapertura 35\*30 tics p\_doors.c:144.

Comprobación: en un nivel con esos specials, cronometrar el cierre a 30 s y la reapertura.

### R9. Una puerta se puede abrir de tres maneras

El sistema deberá permitir abrir una puerta de tres formas: usándola de frente (manual), pisando una línea disparadora, o accionando un interruptor. confianza: alta

Evidencia: manual p\_switch.c:338, al pisar p\_spec.c:545,551,557, interruptor p\_switch.c:403,457,463. La manual actúa sobre la puerta de esa misma pared; las otras dos,

sobre los sectores etiquetados a distancia.

Comprobación: abrir la misma clase de puerta por las tres vías en un nivel de prueba.

### R10. Una bala de arma de bala hace 7, 14 o 21 de daño

El sistema deberá aplicar por bala impactada un daño de uno de los valores del conjunto {7, 14, 21}, elegido al azar, igual para pistola, ametralladora y escopeta. confianza: alta

Evidencia: daño  $7 * (P\_Random() \% 3 + 1)$  p\_pspr.c:633  $P\_Random() \% 3 \in \{0,1,2\}$ ,  $+1 \in \{1,2,3\}$ ,  $\times 7 \rightarrow \{7,14,21\}$ . Las tres armas pasan por P\_GunShot: pistola p\_pspr.c:661, ametralladora p\_pspr.c:753, escopeta con siete perdigones p\_pspr.c:686-687.

Comprobación: instrumentar P\_GunShot o sembrar el generador y registrar el daño; debe caer en {7,14,21}, no en {5,10,15} (los valores de la build pública).

### R11. Cada disparo gasta una bala

El sistema deberá descontar una unidad de munición por disparo; la escopeta gasta una por disparo aunque suelte siete perdigones. confianza: alta

Evidencia: pistola p\_pspr.c:654, escopeta p\_pspr.c:678.

Comprobación: disparar con el contador de munición a la vista y ver que baja de uno en uno.

### R12. Un cargador da 15 balas

El sistema deberá sumar 15 balas al recoger un cargador, hasta un máximo de 200; por encima del tope el cargador no suma. En dificultad baby y nightmare la munición recogida se duplica. confianza: alta

Evidencia: clipammo[am\_clip] = 15 p\_inter.c:59, multiplicado por las cargas del ítem p\_inter.c:91 tope 200 p\_inter.c:58, rechazo al tope p\_inter.c:87-88 doble en baby/nightmare p\_inter.c:95-98.

Comprobación: recoger un cargador con el contador visible y ver el incremento de 15 (no de 10, el valor de la build pública).

## 3. Tabla de parámetros de negocio

Los valores configurables del sistema: cada uno vive en una línea concreta y cambiarlo cambia una regla observable. confianza: alta

| Parámetro                        | Valor        | Unidad              | Regla  | Evidencia     |
|----------------------------------|--------------|---------------------|--------|---------------|
| Espera de la puerta arriba       | 105 (3,00 s) | tics                | R2     | p_spec.h:365  |
| Velocidad de la puerta normal    | 3            | ud. de mapa/tic     | R3     | p_spec.h:364  |
| Factor de las puertas rápidas    | $\times 4$   | multiplicador       | R4     | p_doors.c:487 |
| Hueco bajo el dintel al abrir    | 4            | ud. de mapa         | altura | p_doors.c:498 |
| Espera del cierre programado     | 30           | segundos            | R8     | p_doors.c:521 |
| Espera de la apertura programada | 5            | minutos             | R8     | p_doors.c:549 |
| Multiplicador de daño de bala    | 7            | pts/unidad del dado | R10    | p_pspr.c:633  |
| Balas por cargador               | 15           | balas               | R12    | p_inter.c:59  |
| Tope de balas                    | 200          | balas               | R12    | p_inter.c:58  |

## 4. Historias de usuario y criterios de aceptación

Los criterios se expresan en Gherkin (Dado / Cuando / Entonces), listos para automatizar.

### HU-1. Cruzar una puerta normal

**Como** jugador, **quiero** abrir una puerta usándola **para** pasar al otro lado.

**Dado** que estoy frente a una puerta normal sin cerradura,  
**cuando** pulso «usar»,  
**entonces** la puerta sube hasta abrirse. `p_doors.c:464,168`  
**Dado** que la puerta está del todo abierta,  
**cuando** pasan 105 tics (3,00 s), `p_spec.h:365`  
**entonces** la puerta empieza a cerrarse sola. `p_doors.c:81-85`  
**Dado** que la puerta se está cerrando,  
**cuando** la cruzo por debajo,  
**entonces** la puerta se reabre en lugar de aplastarme. `p_doors.c:159-163`

### HU-2. Puerta con llave

**Como** jugador sin la llave azul, **quiero** una respuesta clara al intentar abrir una puerta azul.

**Dado** que no tengo tarjeta ni calavera azul,  
**cuando** uso una puerta azul (special 26/32),  
**entonces** no se abre y suena el gruñido. `p_doors.c:376-381`  
**Dado** que tengo la llave azul,  
**cuando** uso la puerta azul,  
**entonces** se abre como una normal. `p_doors.c:474`

### HU-3. Puerta rápida

**Como** jugador, **quiero** que las puertas rápidas se noten distintas **para** anticipar su cierre.

**Dado** que uso una puerta rápida (special 117),  
**cuando** sube,  
**entonces** lo hace a cuádruple velocidad `p_doors.c:487` y con su propio sonido de apertura. `p_doors.c:444`  
**Dado** que la puerta rápida se cierra,  
**cuando** llega al suelo,  
**entonces** suena su cierre característico. `p_doors.c:132-133`

### HU-4. Disparar con un arma de bala

**Como** jugador, **quiero** que cada bala haga daño **para** eliminar enemigos.

**Dado** que disparo un arma de bala y el disparo impacta,  
**cuando** se aplica el daño,  
**entonces** es 7, 14 o 21 puntos `p_pspr.c:633`, igual para pistola, ametralladora y escopeta.  
**Dado** que gasto una bala por disparo, `p_pspr.c:654`  
**cuando** recojo un cargador,  
**entonces** recupero 15 balas (tope 200). `p_inter.c:59` `p_inter.c:58`

**Dado** que ya estoy al tope de balas,  
**cuando** piso un cargador,  
**entonces** no lo recojo. `p_inter.c:87-88`

---

## 5. Incertidumbre declarada

Los límites de esta especificación, dichos sin maquillar:

- La **duración total** de apertura (en segundos) no es una regla fija: depende de la altura del hueco (geometría del nivel) dividida por la velocidad. Solo la **espera arriba** (R2) es un tiempo fijo. Confianza baja sobre «cuántos segundos tarda en abrir del todo» sin un nivel concreto.
- El daño de bala (R10) es **aleatorio dentro de un conjunto**, no un valor único: la afirmación correcta es «uno de {7, 14, 21}», no un número fijo por disparo.
- Las puertas programadas (R8) solo se observan si el nivel usa esos disparadores; en un nivel que no los tenga, no hay nada que cronometrar. Confianza media por dependencia del WAD.
- El doble de munición (R12) depende de la dificultad `p_inter.c:95-98`: «15 balas» es la de dificultad normal; en baby y nightmare se observan 30.