

team banzai

BACKLOG DE PRODUCTO (SCRUM)

Puertas y munición en DOOM

Épicas, historias de usuario y criterios de aceptación en Gherkin

Sobre este documento. Este documento presenta el mismo conocimiento que la documentación técnica (arc42) y la funcional (ISO/IEC/IEEE 29148), esta vez como un **backlog de producto Scrum**: tres épicas, doce historias de usuario con el molde «Como / quiero / para», y criterios de aceptación en **Gherkin** (Dado / Cuando / Entonces), listos para automatizar. La prioridad usa **MoSCoW**. La estimación en puntos es **ilustrativa**: mide esfuerzo relativo que acuerda un equipo con su propia velocidad, y aquí no hay tal equipo. Lo que no es ilustrativo son las citas: cada `archivo:línea` entre paréntesis apunta a una línea real de código-mutado/linuxdoom-1.10/, commit `a77dfb9`, comprobable como en las otras documentaciones.

Índice

1. Cómo leer este backlog	2
2. Las tres épicas	2
ÉPICA A · El ciclo de vida de una puerta	2
ÉPICA B · Control de acceso y variantes de puerta	2
ÉPICA C · Armas de bala y economía de munición	2
3. Backlog priorizado	2
4. Historias de usuario con criterios de aceptación	3
HU-01 · Abrir una puerta usándola	3
HU-02 · La puerta se cierra sola tras la espera	3
HU-03 · No morir aplastado bajo una puerta que baja	3
HU-04 · Notar la velocidad de apertura	4
HU-05 · Puertas con llave de color	4
HU-06 · Puertas rápidas de apertura y cierre inmediatos	4
HU-07 · Puertas que se accionan solas por tiempo	5
HU-08 · Abrir la misma puerta por tres vías	5
HU-09 · Los enemigos abren puertas pero no las cierran	6
HU-10 · Cada bala hace un daño variable	6
HU-11 · Un cargador repone munición hasta el tope	6
HU-12 · Cada disparo gasta una bala	7
5. Definición de terminado (Definition of Done)	7
6. Lo que este backlog no cubre (incertidumbre declarada)	7

1. Cómo leer este backlog

El producto documentado es el comportamiento de las **puertas** de un nivel y el **daño y la munición** de las armas de bala. Se organiza en tres épicas. Cada historia sigue el molde «Como [rol] quiero [algo] para [beneficio]» y trae sus criterios de aceptación en Gherkin, listos para automatizar como pruebas. Las conversiones a segundos usan $TICRATE = 35$ tics por segundo `doomdef.h:122`. En los criterios, la referencia `archivo:línea` va como comentario Gherkin (# al final de la línea).

Sobre la estimación. Los puntos de historia miden el esfuerzo relativo que acuerda un equipo concreto con su propia velocidad; aquí sirven para enseñar el formato, no para planificar un sprint. Lo verificable son las citas: cada valor refleja el código a la vista (con las cuatro cifras cambiadas), nunca la build pública de la Doom Wiki.

2. Las tres épicas

ÉPICA A · El ciclo de vida de una puerta

El recorrido completo de una puerta normal: se abre al usarla, sube, espera arriba un tiempo fijo, baja sola y no aplasta a quien la cruza. Es el animador de puertas con sus tres estados $1 \rightarrow 0 \rightarrow -1$ (subiendo, esperando, bajando) `p_doors.c:168,69,118`.

ÉPICA B · Control de acceso y variantes de puerta

Lo que hace que no todas las puertas sean iguales: cerraduras por color de llave, puertas rápidas, puertas que se accionan solas por tiempo, las tres vías de apertura y el trato distinto a los enemigos.

ÉPICA C · Armas de bala y economía de munición

El daño por impacto y la munición de las tres armas de bala (pistola, ametralladora, escopeta): cuánto quita una bala, cuánto gasta un disparo y cuánta munición da un cargador.

3. Backlog priorizado

Vista de tablero: el orden es por épica y prioridad. El detalle de cada historia está en el apartado siguiente.

ID	Historia	Épica	Prioridad	Puntos
HU-01	Abrir una puerta usándola	A	Must	3
HU-02	La puerta se cierra sola tras la espera	A	Must	3
HU-03	No morir aplastado bajo una puerta que baja	A	Should	5
HU-04	Notar la velocidad de apertura	A	Could	2
HU-05	Puertas con llave de color	B	Must	5
HU-06	Puertas rápidas de apertura y cierre inmediatos	B	Should	3
HU-07	Puertas que se accionan solas por tiempo	B	Could	5
HU-08	Abrir la misma puerta por tres vías	B	Should	3

ID	Historia	Épica	Prioridad	Puntos
HU-09	Los enemigos abren puertas pero no las cierran	B	Won't	2
HU-10	Cada bala hace un daño variable	C	Must	3
HU-11	Un cargador repone munición hasta el tope	C	Must	3
HU-12	Cada disparo gasta una bala	C	Should	2

4. Historias de usuario con criterios de aceptación

HU-01 · Abrir una puerta usándola

Como jugador **quiero** abrir una puerta acercándome y pulsando «usar» **para** pasar al otro lado.

Épica A Prioridad: Must 3 puntos

```
Escenario: la puerta sube al usarla
  Dado que estoy frente a una puerta normal sin cerradura
  Cuando pulso la tecla de usar
  Entonces la puerta arranca a subir hacia su apertura # (p_doors.c:464,168)

Escenario: la puerta llega arriba y se queda abierta
  Dado que la puerta va subiendo
  Cuando alcanza el techo de apertura
  Entonces pasa a esperar arriba y arranca su cuenta atras # (p_doors.c:182)
```

HU-02 · La puerta se cierra sola tras la espera

Como jugador **quiero** que la puerta se cierre sola tras un rato **para** no tener que cerrarla a mano.

Épica A Prioridad: Must 3 puntos

```
Escenario: la puerta espera 3,00 segundos arriba
  Dado que la puerta esta del todo abierta
  Cuando pasan 105 tics de espera # VDOORWAIT=105 (p_spec.h:365), topwait
  (p_doors.c:466)
  Entonces empieza a cerrarse sola con su sonido # (p_doors.c:81-85)
  Y 105 tics / 35 por segundo son 3,00 s # (doomdef.h:122)

Escenario: cronometrar la espera para verificarla
  Dado el juego corriendo con esta copia del codigo
  Cuando cronometro desde que abre hasta que arranca a bajar
  Entonces mido unos 3,0 s, no 4,3 s # la build publica seria 150 tics
```

HU-03 · No morir aplastado bajo una puerta que baja

Como jugador **quiero** que la puerta que baja se reabra si me pilla debajo **para** no morir por descuido.

Épica A Prioridad: Should 5 puntos

Escenario: la puerta se reabre al atrapar algo debajo
Dado que una puerta normal se esta cerrando
Cuando algo la cruza por debajo
Entonces invierte y vuelve a subir sin causar dano # (p_doors.c:151-164)

Escenario: las puertas de solo cierre no se reabren
Dado que la puerta es de tipo close o blazeClose
Cuando algo la cruza por debajo mientras baja
Entonces insiste en cerrar en lugar de reabrirse # (p_doors.c:151-164)

HU-04 · Notar la velocidad de apertura

Como jugador **quiero** que la puerta se mueva a una velocidad perceptible **para** anticipar cuándo estará abierta.

Épica A Prioridad: Could 2 puntos

Escenario: la puerta normal se mueve a 3 unidades por tic
Dado que una puerta normal esta subiendo o bajando
Cuando el animador la mueve un tic
Entonces avanza 3 unidades de mapa # VD0ORSPEED=FRACUNIT*3 (p_spec.h:364),
(p_doors.c:171)

HU-05 · Puertas con llave de color

Como jugador sin la llave **quiero** una respuesta clara al intentar abrir una puerta de color **para** entender que me falta la llave.

Épica B Prioridad: Must 5 puntos

Escenario: sin la llave, la puerta no abre
Dado que no llevo la tarjeta ni la calavera azul
Cuando uso una puerta azul con cerradura
Entonces no se abre y suena un grunido de negativa # azul (p_doors.c:376-381)

Escenario: con la llave, abre como una normal
Dado que llevo la llave del color correcto
Cuando uso la puerta con cerradura
Entonces se abre igual que una puerta normal # (p_doors.c:474)

Escenario: cada color admite tarjeta o calavera
Dado que existe una puerta amarilla o roja con cerradura
Cuando uso la tarjeta o la calavera de ese color
Entonces cualquiera de las dos vale # amarilla (p_doors.c:389-395), roja
(p_doors.c:403-408)

HU-06 · Puertas rápidas de apertura y cierre inmediatos

Como jugador **quiero** que las puertas rápidas se noten distintas **para** no quedarme debajo cuando se cierran de golpe.

Épica B Prioridad: Should 3 puntos

Escenario: la puerta rapida va a cuádruple velocidad
Dado que uso una puerta rapida (special 117 o 118)
Cuando sube
Entonces lo hace a cuatro veces la velocidad normal # VD00RSPEED*4
(p_doors.c:487,492)
Y suena con su propio sonido de apertura # (p_doors.c:444)

Escenario: la puerta rapida suena distinto al cerrar
Dado que una puerta rapida se esta cerrando
Cuando llega al suelo
Entonces suena su cierre caracteristico # (p_doors.c:132-133)

HU-07 · Puertas que se accionan solas por tiempo

Como diseñador de nivel **quiero** puertas que se abran o cierren solas por tiempo **para** montar trampas y atajos temporizados.

Épica B

Prioridad: Could

5 puntos

Escenario: puerta que se cierra a los 30 segundos
Dado un sector con el disparador de cierre programado
Cuando pasan 30 segundos
Entonces la puerta se cierra sin que el jugador intervenga # (p_doors.c:521)

Escenario: puerta que se abre a los 5 minutos
Dado un sector con el disparador de apertura programada
Cuando pasan 5 minutos
Entonces la puerta se abre sola # (p_doors.c:549)

Escenario: puerta que se cierra, espera abajo y reabre
Dado una puerta de tipo close30Then0pen
Cuando se cierra
Entonces espera 30 segundos abajo antes de reabrirse # 35*30 tics
(p_doors.c:144)

HU-08 · Abrir la misma puerta por tres vías

Como jugador **quiero** poder abrir una puerta de varias formas **para** resolver el nivel según lo que tenga a mano.

Épica B

Prioridad: Should

3 puntos

Escenario: abrir usandola de frente
Dado que estoy frente a una puerta manual
Cuando pulso usar
Entonces se abre la puerta de esa misma pared # (p_switch.c:338)

Escenario: abrir pisando una linea disparadora
Dado que hay una puerta remota con linea disparadora
Cuando piso esa linea
Entonces se abre el sector etiquetado a distancia # (p_spec.c:545,551,557)

Escenario: abrir con un interruptor
Dado que hay un interruptor asociado a una puerta remota

```
Cuando lo acciono
Entonces se abre el sector etiquetado # (p_switch.c:403,457,463)
```

HU-09 · Los enemigos abren puertas pero no las cierran

Como jugador quiero que los enemigos no puedan cerrarme una puerta encima **para** que la mecánica sea justa.

Épica B Prioridad: Won't esta iteración 2 puntos

```
Escenario: un enemigo cruza pero no cierra
Dado una puerta ya activa que un enemigo esta usando
Cuando el enemigo la cruza
Entonces no se fuerza el cierre: quien la usa no es el jugador #
(p_doors.c:430-431)
```

HU-10 · Cada bala hace un daño variable

Como jugador quiero que cada bala haga daño **para** eliminar enemigos, con algo de variación por disparo.

Épica C Prioridad: Must 3 puntos

```
Escenario: el dano de una bala cae en un conjunto de tres valores
Dado que disparo un arma de bala y el disparo impacta
Cuando se aplica el dano
Entonces es 7, 14 o 21 puntos, al azar # 7*(P_Random()%3+1) (p_pspr.c:633)

Escenario: las tres armas de bala usan la misma regla
Dado que disparo la pistola, la ametralladora o la escopeta
Cuando el perdigon impacta
Entonces todas aplican {7, 14, 21} # pistola (p_pspr.c:661), escopeta
(p_pspr.c:686-687)
```

HU-11 · Un cargador repone munición hasta el tope

Como jugador quiero recoger cargadores **para** seguir disparando, sin pasarme del máximo que puedo llevar.

Épica C Prioridad: Must 3 puntos

```
Escenario: recoger un cargador suma balas
Dado que tengo sitio para mas balas
Cuando recojo un cargador
Entonces gano 15 balas # clipammo[am_clip]=15 (p_inter.c:59)

Escenario: no recoger por encima del tope
Dado que ya llevo el maximo de 200 balas # tope (p_inter.c:58)
Cuando piso un cargador
Entonces no lo recojo # rechazo al tope (p_inter.c:87-88)
```

HU-12 · Cada disparo gasta una bala

Como jugador **quiero** que cada disparo consuma munición **para** tener que administrar las balas.

Épica C

Prioridad: Should

2 puntos

```
Escenario: la pistola gasta una bala por disparo
  Dado que disparo la pistola
  Cuando sale el disparo
  Entonces se descuenta una unidad de municion  # (p_pspr.c:654)

Escenario: la escopeta gasta una bala aunque suelte siete perdigones
  Dado que disparo la escopeta
  Cuando sale el disparo de siete perdigones
  Entonces se descuenta una sola unidad  # (p_pspr.c:678)
```

5. Definición de terminado (Definition of Done)

Una historia se da por terminada cuando: sus criterios Gherkin pasan como pruebas automatizadas; cada valor citado se ha comprobado a mano en su `archivo:línea` y el comportamiento se ha visto en el juego corriendo desde esta copia del código, no solo leído.

6. Lo que este backlog no cubre (incertidumbre declarada)

- La **duración total** de apertura de una puerta en segundos no es una historia estimable sin un nivel concreto: depende de la altura del hueco (geometría del WAD) dividida por la velocidad. Solo la **espera arriba** (HU-02) es un tiempo fijo y citable.
- El **daño de bala** (HU-10) es aleatorio dentro de un conjunto: el criterio correcto es «uno de 7, 14 o 21», no un número fijo por disparo. Automatizarlo exige sembrar el generador de azar.
- Las **puertas por tiempo** (HU-07) solo se observan si el nivel usa esos disparadores; por eso la historia es Could, no Must.
- El doble de munición en dificultad baby y nightmare `p_inter.c:95-98` queda fuera de HU-11: «15 balas» es la cifra de dificultad normal.