



QA PARA INDIES

**Deja de apagar fuegos y
empieza a prevenir bugs**

¿De qué voy a hablar?

1. ¿Cómo se hace QA?
2. ¿Cómo se hacen tests?
3. ¿Cómo se automatiza?





¿Quién soy yo?

- Nepo (elle/they)
- QA (8 años), freelance (1 año 😊)
- “Configuración de CI para jams”
- Nokorpo: consultoría, juegos por encargo



Parte 1

¿Cómo se hace QA?



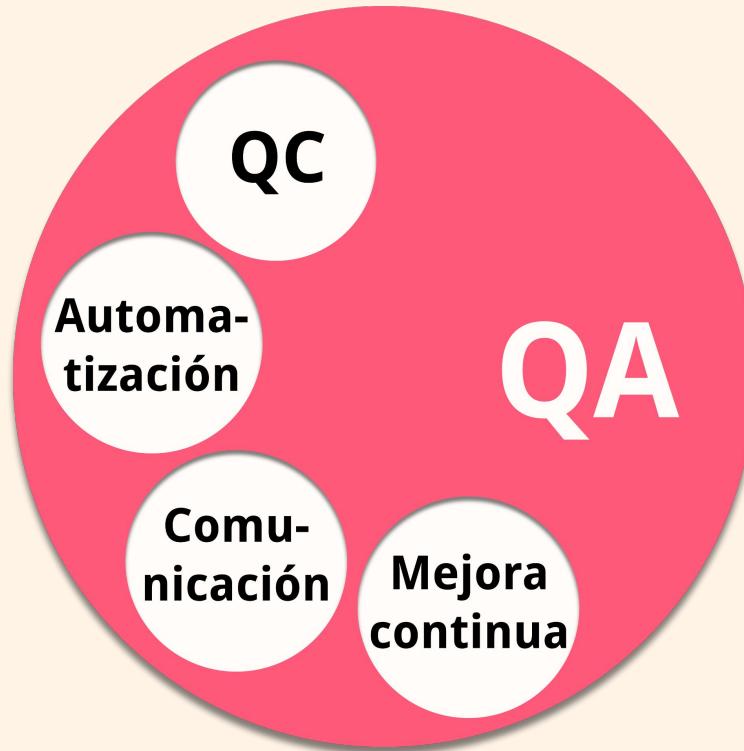
¿Qué significa QA?



QA ≠ QC



QA ≠ QC



¿Qué es QA?

Ideación

- Brainstorm
- Investigar
- Prototipar

Preproducción

- GDD
- Vertical slice
- Plan de desarrollo

Producción

- Implementar
- Iterar
- Iterar
- Iterar
- ...

Postproducción

- Balancing
- Bug fixing
- Polishing

Release

- Build
- Testing
- Marketing

Live Players



¿Qué es QA?

Ideación

Preproducción

Producción

Postproducción

Release

Live Players

QA



¿Qué es QA?

Ideación

Desarrollo

Validación

Postproducción

Release

Live Play

QA



¿Qué es QA?

Ideación

Preproducción

Producción

Postproducción

Release

Live Players

QA

QA

QA
QA
QA
QA

QA

QA

QA

QA trabaja sobre **procesos** del equipo que se dan a lo largo del desarrollo



Estrategia de QA: objetivo

Objetivo: ¿encontrar bugs?

Sí, pero...



Estrategia de QA: objetivo

Ideación

Preproducción

Producción

Postproducción

Release

Live Players



Estrategia de QA: objetivo

Ideación

Preproducción

Producción

Postproducción

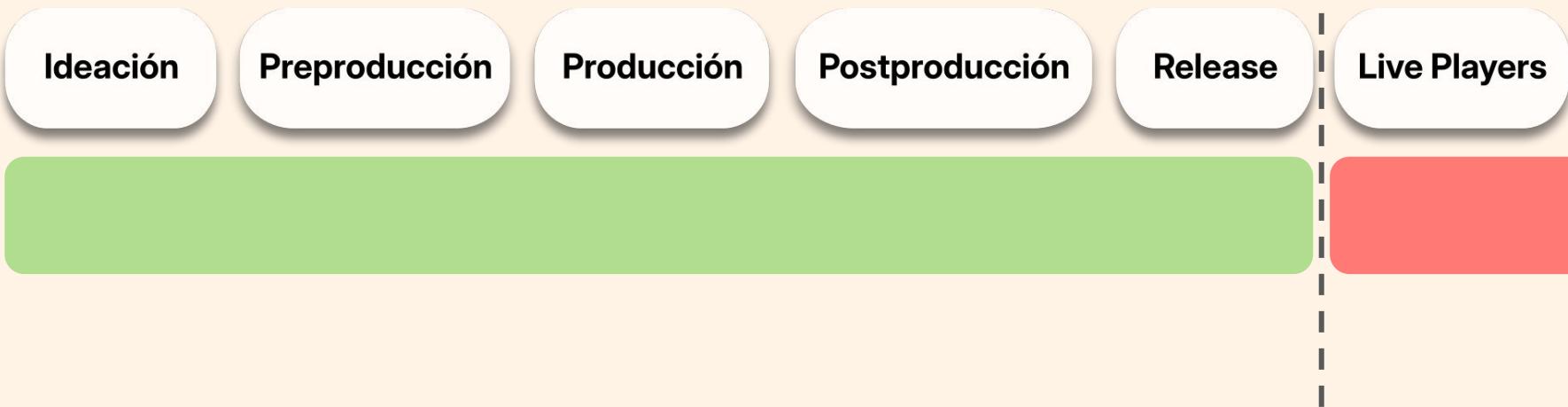
Release

Live Players

A partir de aquí los
bugs afectan a
jugadores



Estrategia de QA: objetivo



- Objetivo: encontrar bugs **antes de la release**



Estrategia de QA: objetivo

Ideación

Preproducción

Producción

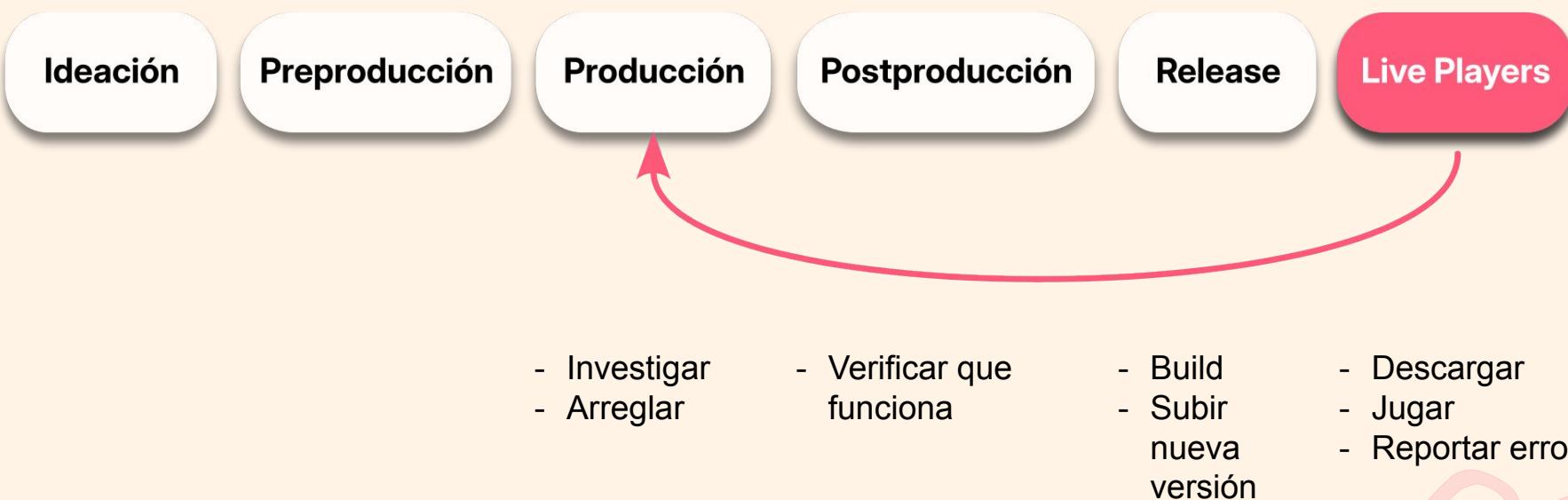
Postproducción

Release

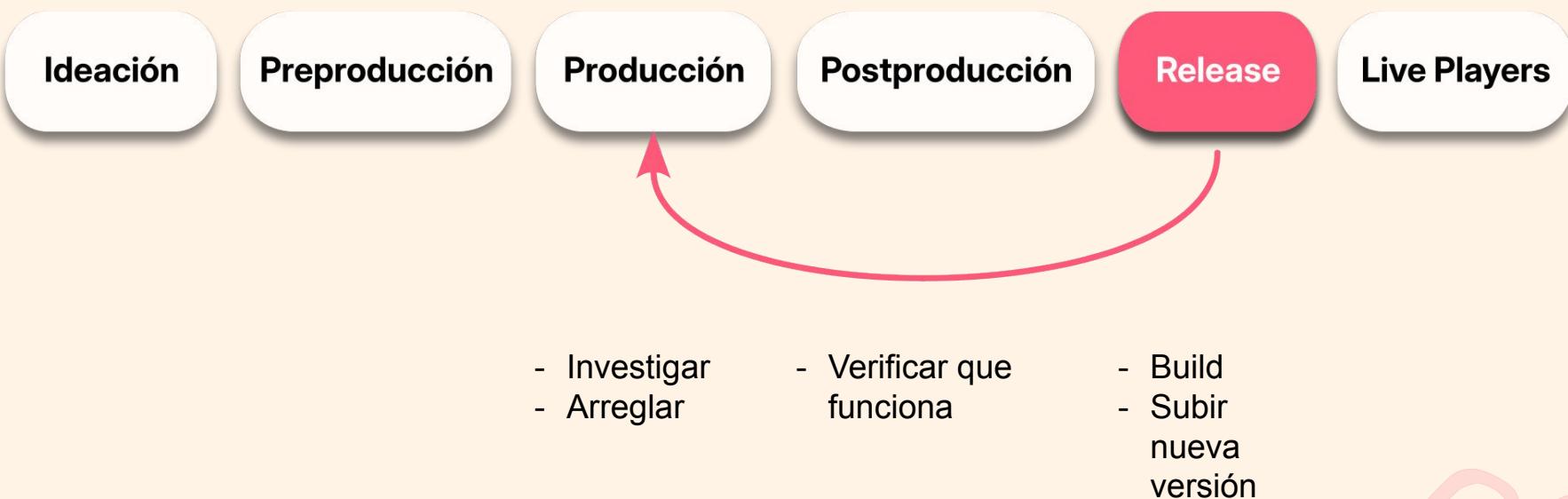
Live Players



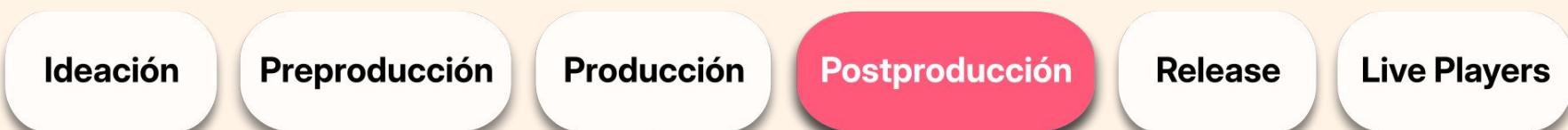
Estrategia de QA: objetivo



Estrategia de QA: objetivo



Estrategia de QA: objetivo



- Investigar
- Arreglar
- Verificar que funciona



Estrategia de QA: objetivo



- Investigar
- Arreglar



Estrategia de QA: objetivo

Ideación

Preproducción

Producción

Postproducción

Release

Live Players



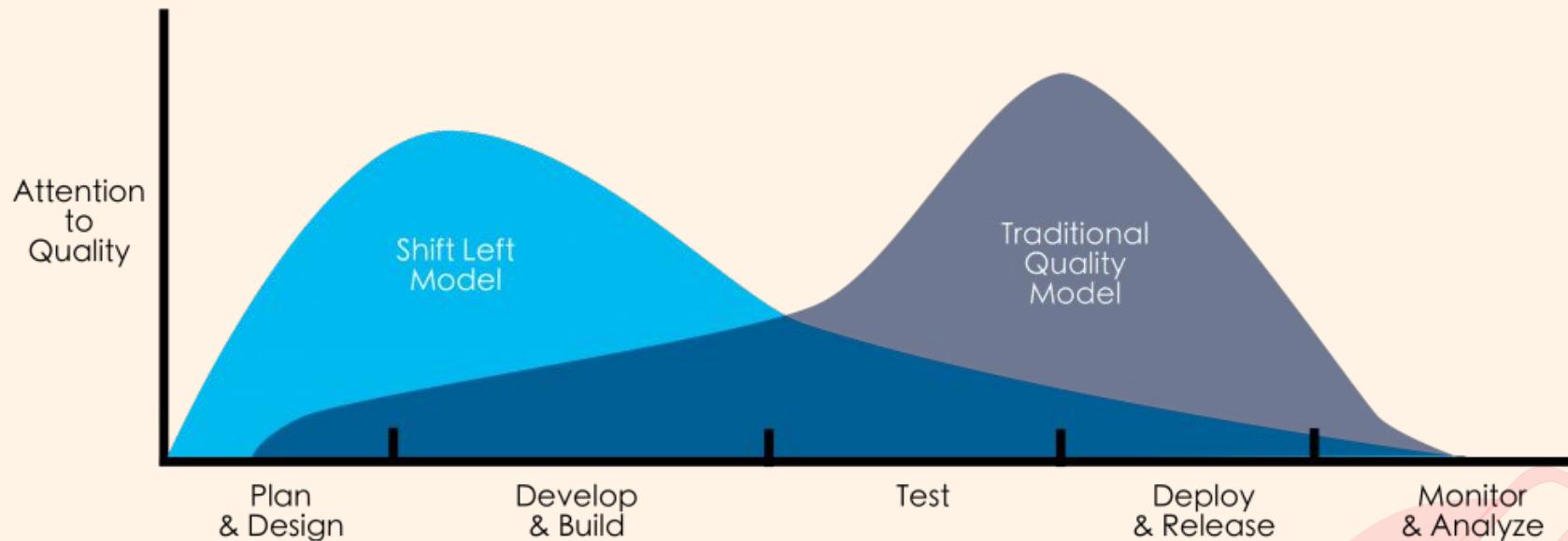
Estrategia de QA: objetivo



- Objetivo: encontrar bugs **lo antes posible**



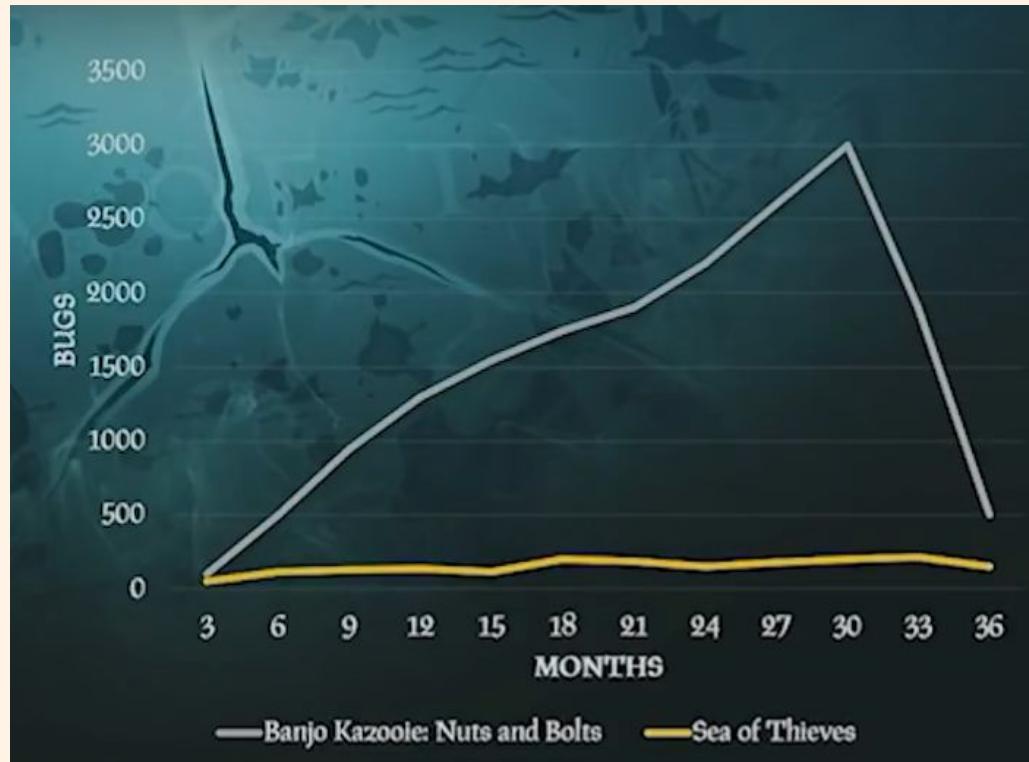
Shift left testing



"It's 2017: Test automation is not optional when building mobile apps!" — Geert van der Cruijsen

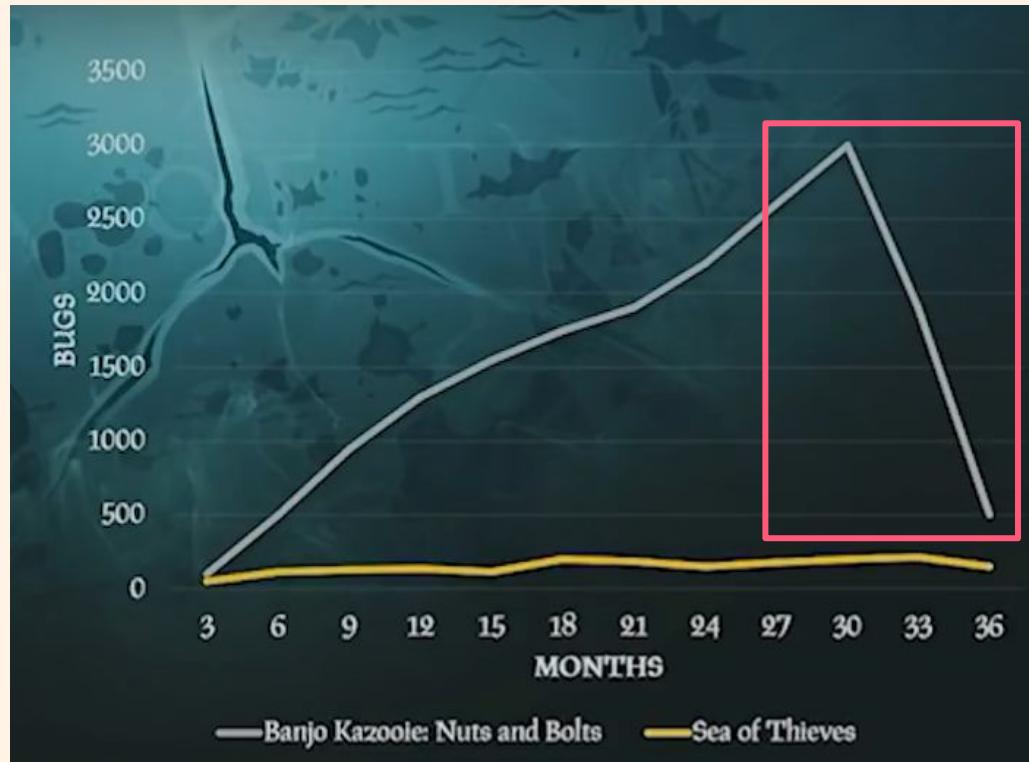


Shift left testing



"Automated Testing of Gameplay Features in 'Sea of Thieves'", Robert Masella, GDC

Shift left testing



iCrunch!

"Automated Testing of Gameplay Features in 'Sea of Thieves'", Robert Masella, GDC

¿Qué es un proceso?

Ideación

- Brainstorm
- Investigar
- Prototipar

Preproducción

- GDD
- Vertical slice
- Plan de desarrollo

Producción

- Implementar
- Iterar
- Iterar
- Iterar
- ...

Postproducción

- Balancing
- Bug fixing
- Polishing

Release

- Build
- Testing
- Marketing

Live Players



¿Qué es un proceso?

"Una serie de tareas realizadas con el fin de alcanzar un **objetivo**"

- Hacer código/arte e **integrarlo** en el juego
- Subir una build a Steam y **lanzarla** al público
- Tener una reunión para **tomar decisiones**
- Hacer un brainstorming para **buscar ideas**
- ... y muchos más ejemplos



Testing de procesos

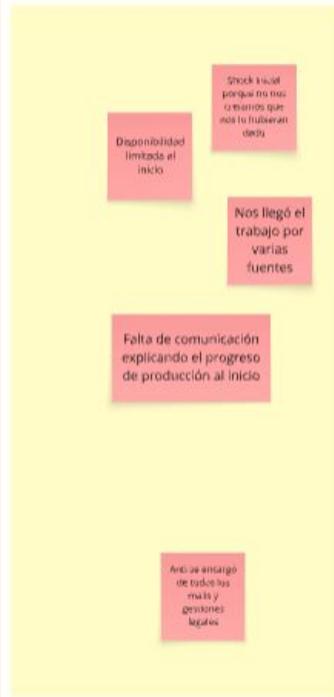
La idea es muy bonita, pero...

- ¿Cómo se testea **un proceso**?
- ¿Se puede testear **una reunión**?

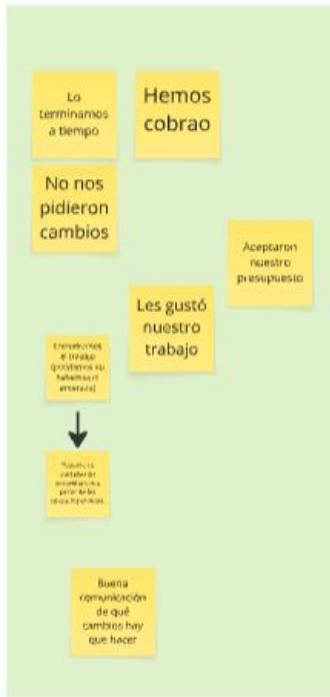


Post-mortem

What happened



What went well



What didn't go well



How will we adjust in the future



Retrospectiva

Retro (30')

Desde 27
Mar

Cada columna representa una categoría. Tenemos que pensar en nuestra semana bajo esas categorías y compartir lo que nos ha afectado. Tendremos 5-10 minutos para escribir lo que se nos ocurra, y el resto del tiempo para discutir cómo evitar las cosas malas y fomentar las buenas.

Good



Bad



Real



Agreements

Documentar las ideas que se nos vayan ocurriendo en el Drive - Nepo creará el doc

Experimento:
entrar en reunión cuando estemos currando (aunque sea mutado)

Actions



ROTI



Espacio reservado para dejar dudas, consejos, peticiones o amenazas de muerte.

Duro

detallado

Retrospectiva

Retro (30')

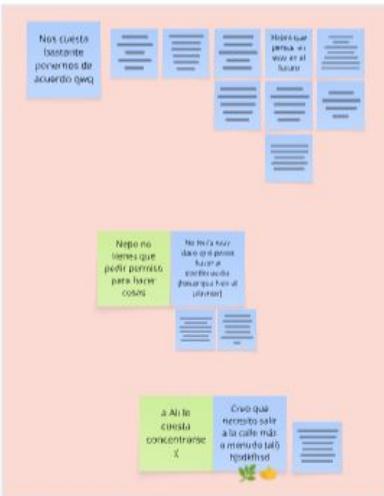
Desde 27 Mar

Cada columna representa una categoría. Tenemos que pensar en nuestra semana bajo esas categorías y compartir lo que nos ha afectado. Tendremos 5-10 minutos para escribir lo que se nos ocurra, y el resto del tiempo para discutir cómo evitar las cosas malas y fomentar las buenas.

Good



Bad



Real

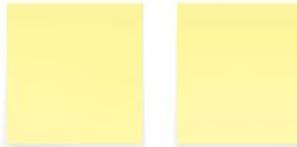


Agreements

Documentar las ideas que se nos vayan ocurriendo cuando estemos currando en el Drive - Nepo creará el doc

Experimento: entrar en reunión cuando estemos currando (aunque sea mutado)

Actions



ROTI



Espacio reservado para dejar dudas, consejos, peticiones o amenazas de muerte.



Duró demasiado

Pedir feedback

Espacio al final de una reunión para dejar feedback



Espacio reservado para dejar dudas, consejos, peticiones o amenazas de muerte.

La persona que esté hablando de un post-it puede decir "vale, siguiente cosa"

Duró demasiado

Iterar proceso

¿Cómo se testea una reunión/proceso?

- Echando la **vista atrás**
- Pensando en qué **cambiar para mejorar**

Iterar desbloquea la **mejora continua**



¿Y esto funciona?

Brainstorming Proyecto A:

- Sin guía/estructura
- 4h discutiendo sin una conclusión. Otro día 2h más para acordar una idea

Brainstorming Proyecto B:

- Sesión con dos partes: 1) lanzar ideas, 2) proponer juegos con esas ideas
- Listo en 2h



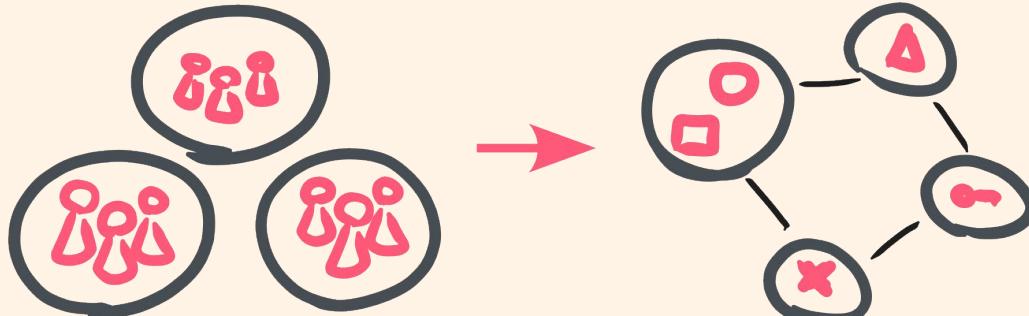
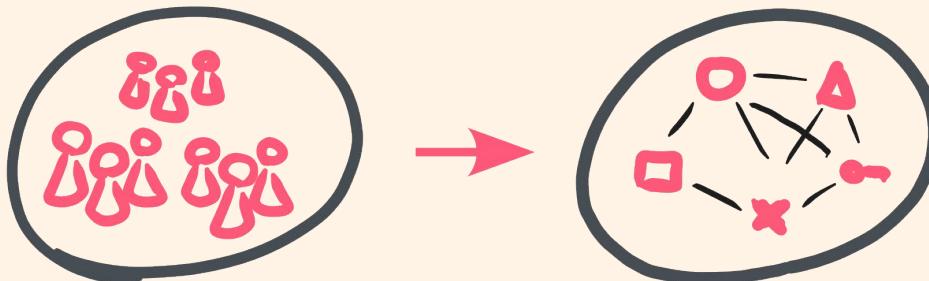
Ley de Conway

“Las organizaciones que diseñan sistemas (...) están obligadas a producir diseños que son copias de sus estructuras de comunicación.”

— Melvin Conway



Ley de Conway



Resumen

- Objetivo de QA: encontrar/prevenir errores **lo antes posible**
- Reducir **feedback loops**
- Podemos aplicar QA sobre **todo el proceso de desarrollo**
- Echar la vista atrás (**bueno, malo**) y pensar en **qué cambiar**

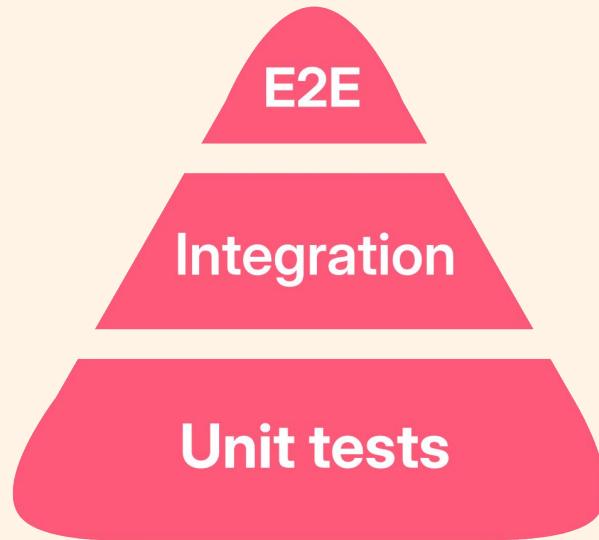


Parte 2

¿Cómo se hacen tests?



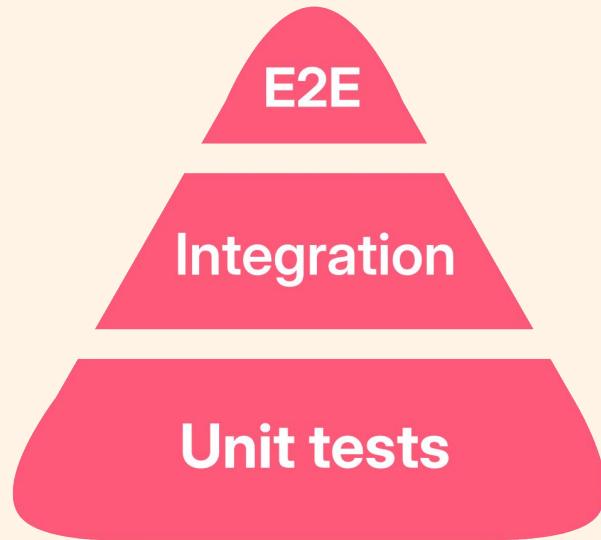
Implementación de pruebas



Pyramid testing model, Mike Cohn & Lisa Crispin



Implementación de pruebas

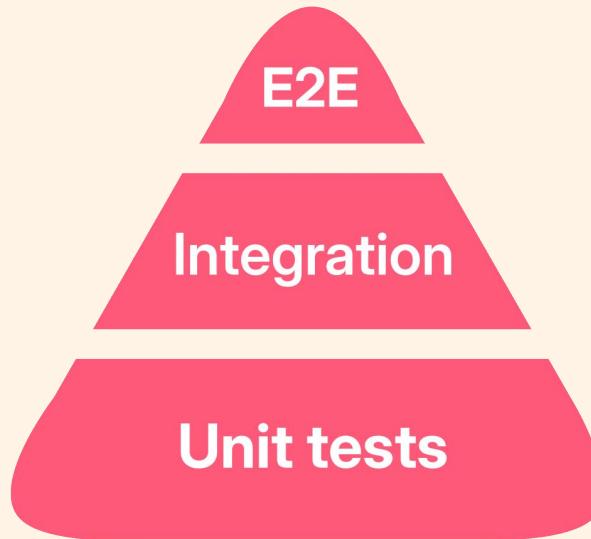


Pyramid testing model, Mike Cohn & Lisa Crispin

Unit test vs. Integration test



Implementación de pruebas



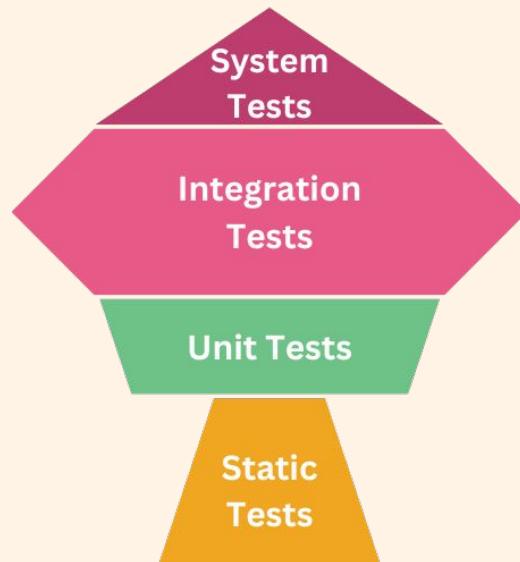
Pyramid testing model, Mike Cohn & Lisa Crispin

- Antiguo
- Huele a waterfall
- ¿Unit tests pa qué?

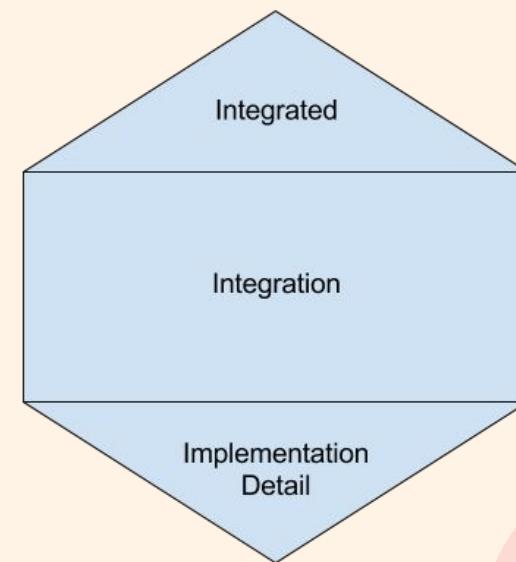


Implementación de pruebas

"Write tests. Not too many. Mostly integration" — Guillermo Rauch



Trophy testing model, Kent C. Dodds

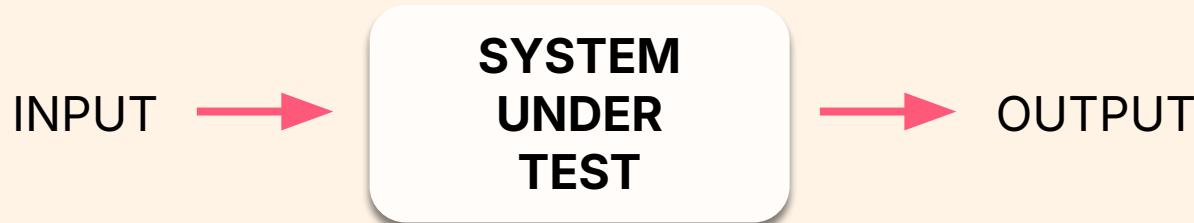


Microservices honeycomb testing model, Spotify



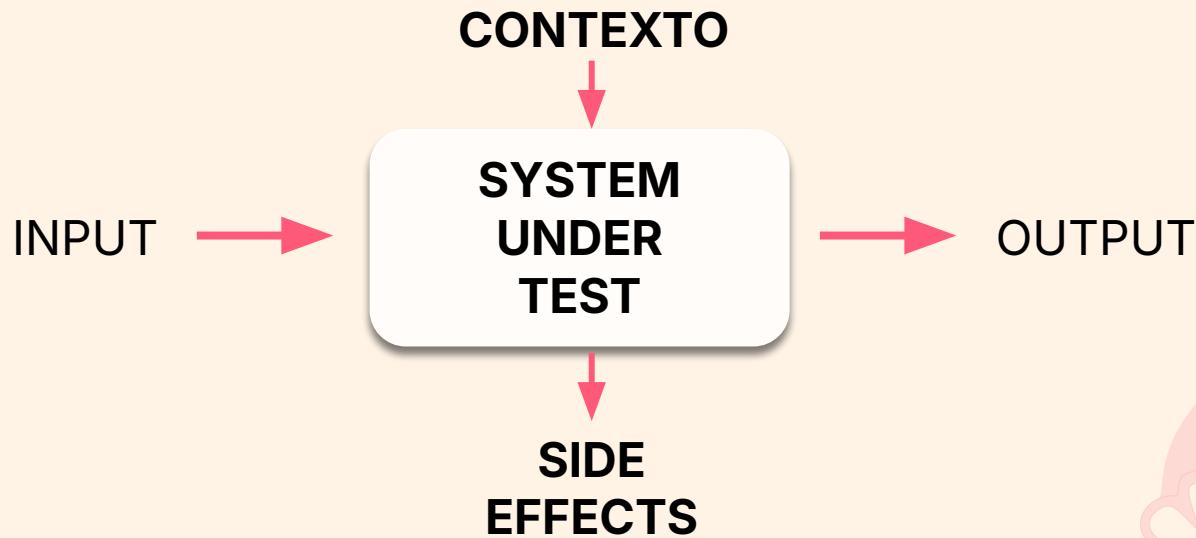
¿Por qué integración?

- ¿Cuándo fue la última vez que escribiste **una unidad** de código?
- ¿O una **función pura**?



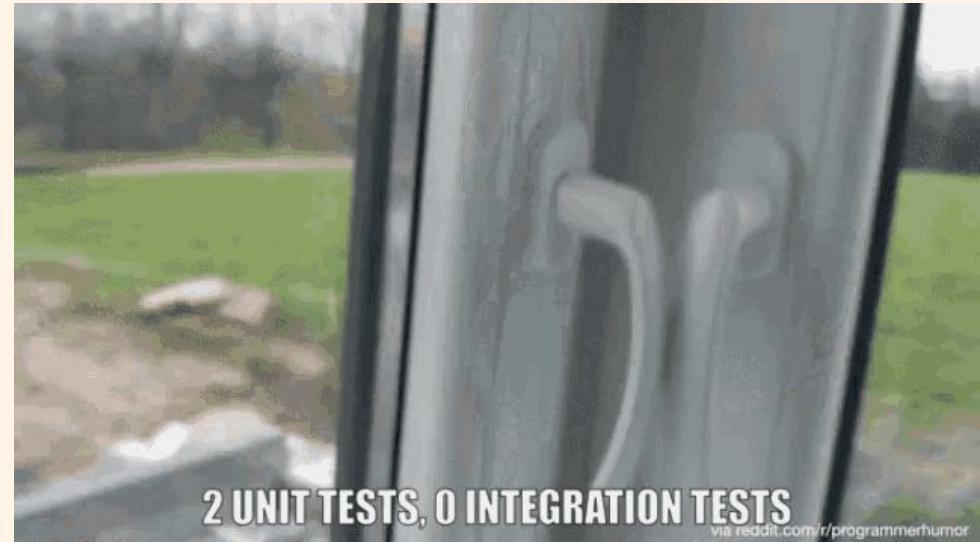
¿Por qué integración?

- El código que escribimos está **integrado** con un **motor, librerías...**
- El **estado** de estos sistemas les afectan, y **generan cambios** en él



¿Por qué integración?

- Si no testeamos **el conjunto**, no nos aseguramos de que **funciona**
- *Tenemos que hacer tests de integración*



¿Cómo hago tests de integración?

GIVEN

Preparar la **escena**

WHEN

Actuar sobre ella, llamar al método

THEN

Hacer comprobaciones, **asserts**



[empty] +
File Edit Search Go To Debug

Online Docs Search Help

Filter Scripts



- character_selection.gd
- main_menu.gd
- main_menu_options.gd
- player_preview.gd
- scene_manager.gd
- smoke_test.gd
- test_game_rules.gd

scene_manager.gd



Filter Methods



- _ready
- change_scene
- persist

```
1  extends Node
2  ## Manages scene transitions
3
4  @onready var curtain: Curtain = $UI/Curtain
5
6  func _ready() -> void:
7      EventBus.change_scene.connect(change_scene)
8
9  ## Change scenes with the curtain animation
10 func change_scene(event: EventBus.ChangeSceneEvent) -> void:
11     curtain.play_animation()
12     await curtain.change_scene_now
13
14 for child in $CurrentScene.get_children():
15     child.queue_free()
```

100% 6 : 16 Tabs

▶ Run All Current: ▶ test_game_rules.gd → ▶ test_scoreboard_ui_shown_on_player_death

Passing 19.0	Failing 0.0	Pending 0.0	Orphans 0.0	Errors 0.0	Warnings 0.0
--------------	-------------	-------------	-------------	------------	--------------

+	+	All: +	+	Sync: <input type="button" value="T"/>
---	---	--------	---	--

Everything passed!



Copy Clear

¡Más tipos de tests!

- Tests para **problemas específicos**
- Encuentra el problema y cúbrello con un test
- Evita que algo malo ocurra **sin que lo sepas**



Screenshot test

Compara screenshots para encontrar diferencias

List of items:

- Item 1
- Item 2
- Item 3
- Item 4

CAPTURA ORIGINAL



Screenshot test

¿Cuál es la diferencia entre estas 2 imágenes?

- List of items:
- Item 1
 - Item 2
 - Item 3
 - Item 4



CAPTURA ORIGINAL

- List of items:
- Item 1
 - Item 2
 - Item 3
 - Item 4
 - Item 5

CAPTURA DEL TEST



Screenshot test

¿Cuál es la diferencia entre estas 2 imágenes?

- List of items:
- Item 1
 - Item 2
 - Item 3
 - Item 4



CAPTURA ORIGINAL

- List of items:
- Item 1
 - Item 2
 - Item 3
 - **Item 4**
 - Item 5

CAPTURA DEL TEST

■ No existe en la nueva

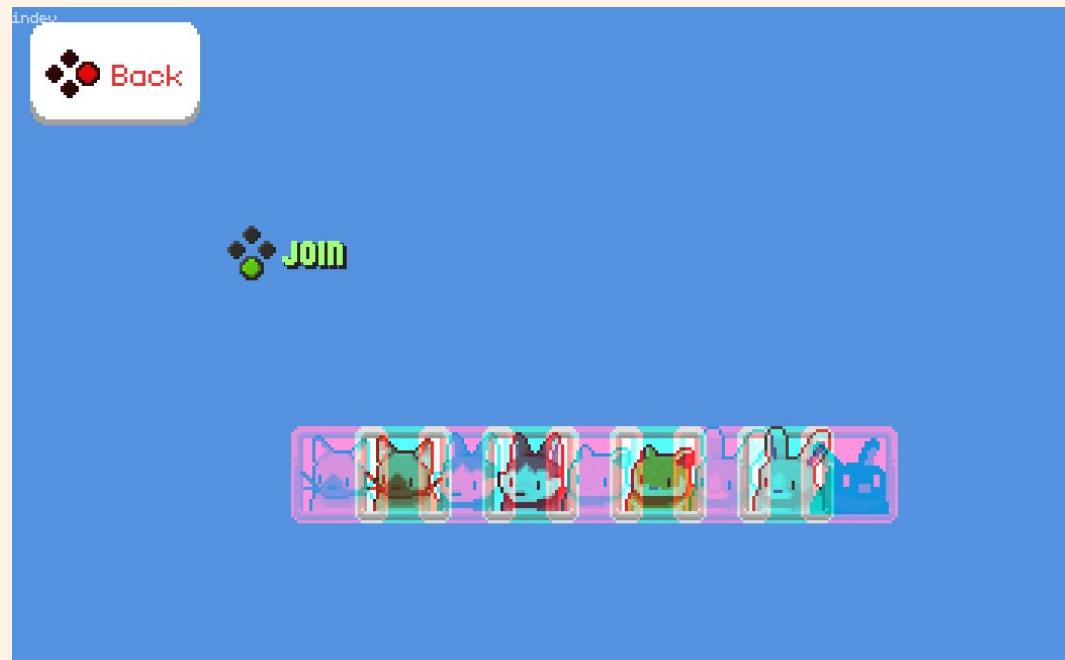
■ Existe en la nueva



Screenshot test

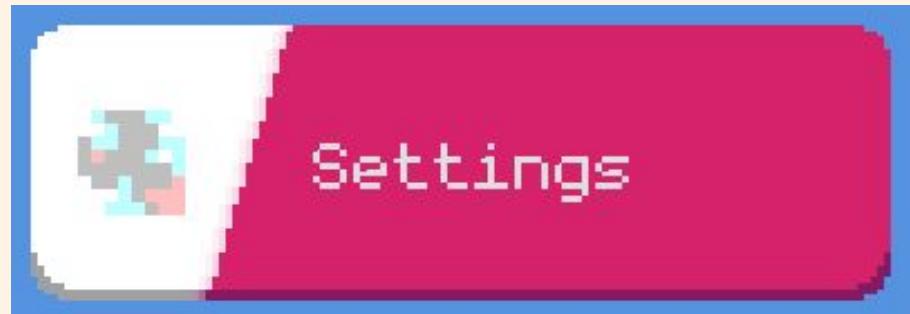
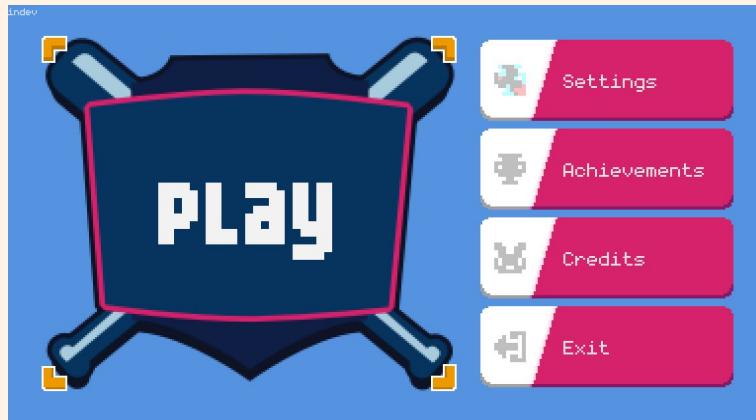
¿Para qué usarlo?

- Pruebas de la UI
- Pruebas de texturas
- Pruebas de shaders



Pantalla de selección de personaje, Michiball

Screenshot test



Menú principal, Michiball

The screenshot shows a screenshot testing interface. At the top, there are numerical values for various metrics: Passing 0.0, Failing 1.0, Pending 0.0, Orphans 1.0, Errors 0.0, and Warnings 0.0. Below these are buttons for filtering by status (Passing, Failing, Pending, Orphans, Errors, Warnings) and sync settings. A tree view shows a test file structure:

- res://src/tests/screenshot/test_menus.gd
 - test_main_menu_screen
 - fail: [0.00110918209877] expected to be < than [0.001]: snapshot is different at line 32

A pink decorative graphic of a hand holding a stylus is visible on the right side of the interface.

Scene Import

Filter: name, tt

- MainMenuOptions
- HBoxContainer
 - Play
- VBoxContainer
- Settings
 - Icon
- Achievements
 - Icon
- Credits
 - Icon
- Exit
 - Icon

FileSystem

res://src/core/game_men

Filter Files

- roster_character.gd
- roster_character.tscn
- roster_container.gd
- team_selection.gd
- credits
- customize_game_sele...
- game_mode_selection
- main_menu_options
- playlist_selection
- settings
- team_selection
- fov_dynamic_camera...
- main_menu.gd

File Edit Search Go To Debug

Filter Scripts

- character_selection.gd
- main_menu.gd
- main_menu_options.gd
- player_preview.gd
- test_menus.gd

Filter Methods

- before_all
- take_screenshot
- test_title_screen
- test_main_menu_screen
- test_settings_general_sc...
- test_settings_audio_scre...
- test_settings_video_scre...
- test_settings_mods_scre...

```

18 func test_title_screen() -> void:
19     var title_scene: PackedScene = load("res://src/core/game_menu/title_screen.tscn")
20     var _title: Node = add_child_autofree(title_scene.instantiate())
21     await(wait_frames(4, "wait para que cargue la escena"))
22
23     var result: ComparisonResult = take_screenshot("title")
24     assert_lt(result.difference_by_percent, THRESHOLD, "snapshot is different")
25
26 func test_main_menu_screen() -> void:
27     var main_menu_node = load("res://src/core/game_menu/main_menu_options/main_menu_options.tscn")
28     var _main_menu: Node = add_child_autofree(main_menu_node)
29     await(wait_frames(4, "wait para que cargue la escena"))
30
31     var result: ComparisonResult = take_screenshot("main_menu")
32     assert_lt(result.difference_by_percent, THRESHOLD, "snapshot is %f different" % result.difference_by_percent)
33
34 func test_settings_general_screen() -> void:
35     var settings_scene: PackedScene = load("res://src/core/game_menu/settings/settings.tscn")
36     var _settings: Node = add_child_autofree(settings_scene.instantiate())
37     await(wait_frames(4, "wait para que cargue la escena"))
38

```

Run All Current: test_menus.gd

●	Passing	1.0	Failing	0.0	Pending	0.0	Orphans	0.0	Errors	0.0	Warnings	0.0
+	All:	+	+	+	+	+	+	+	+	+	+	
+	Passing		Sync:	Sync	T							

Everything passed!

Ran Scripts matching "test_menus.gd"
 Ran Tests matching "test_main_menu_screen"
 ---- All tests passed! ----

Inspector Node History

Icon

Filter Properties

TextureRect

Texture

Expand Mode Keep Size

Stretch Mode Scale

Flip H On

Flip V On

Control

This node is a child of a regular control.

Clip Contents On

Custom Min x 16.0 px

y 16.0 px

Layout Direction Inherited

Layout Mode Anchors

Anchors Pre Center Le (2 changes)

Transform

Localization

Tooltip

Focus

Mouse

Filter Stop

Force Pass Screen On

Default Cursor Arrow

Input Theme

CanvasItem

Screenshot test

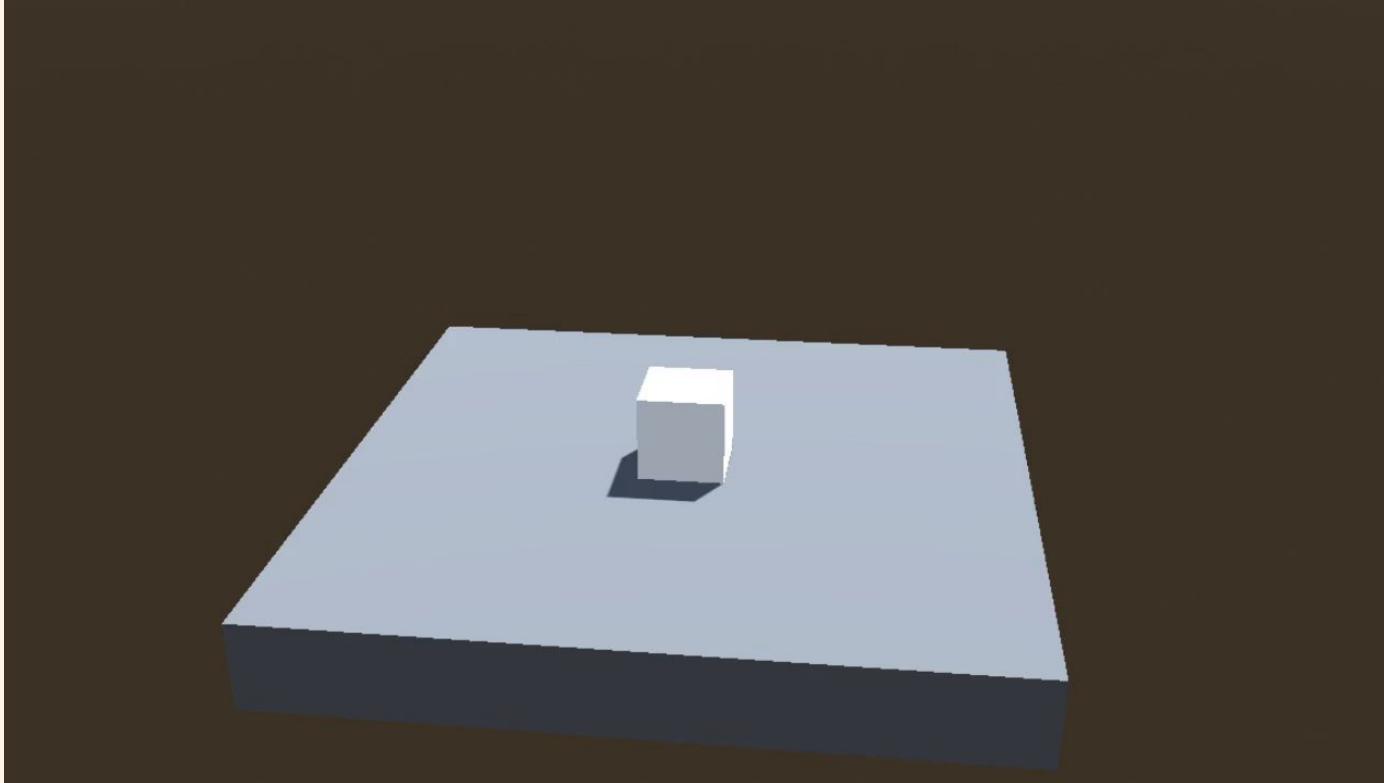
Lo tenemos como un plugin de Godot de código abierto y 100% gratis



<https://github.com/Nokorpo/GDsnap>



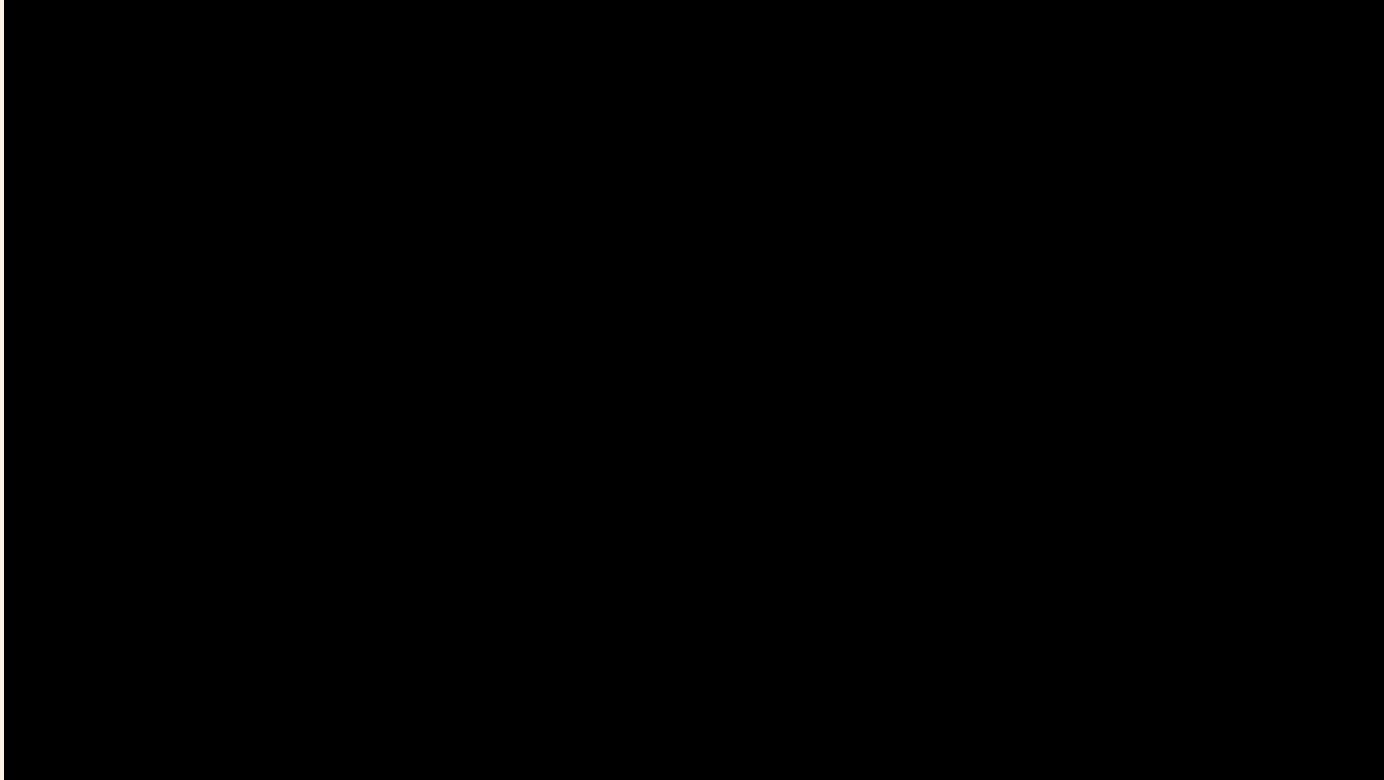
Performance test/benchmark



<https://github.com/Nokorpo/GDsnap>



Performance test/benchmark

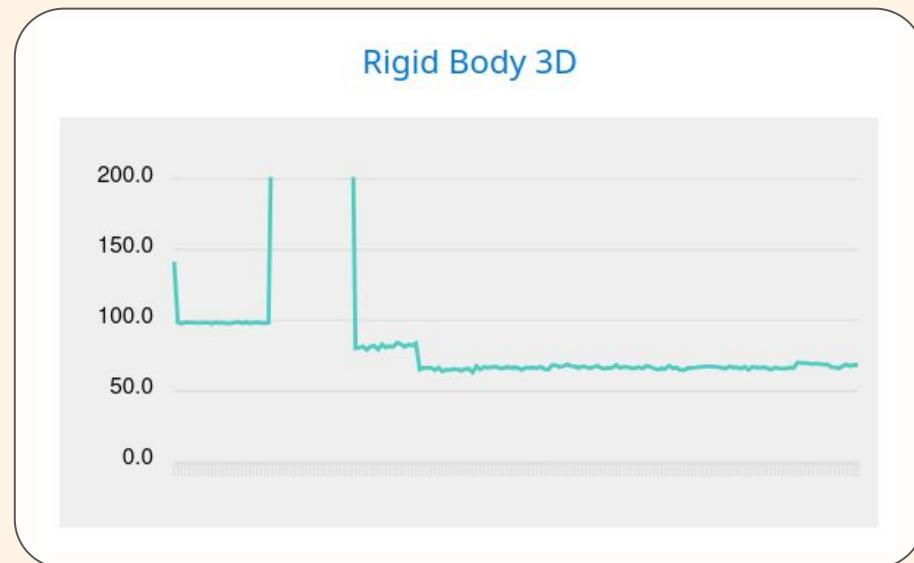


<https://github.com/Nokorpo/GDsnap>



Performance test/benchmark

Ej. <https://benchmarks.godotengine.org/>



Benchmarks de Godot Engine



Performance test/benchmark

¿Para qué usarlo?

- Evolución del uso de recursos de un nivel



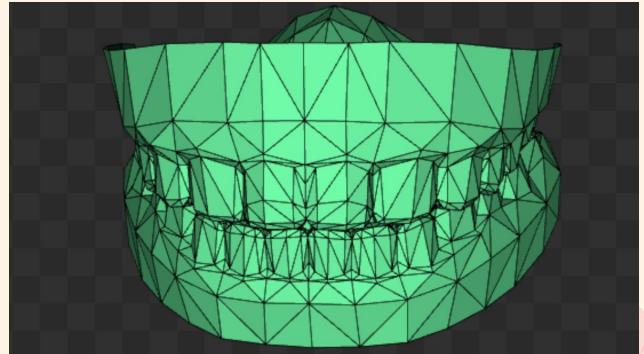
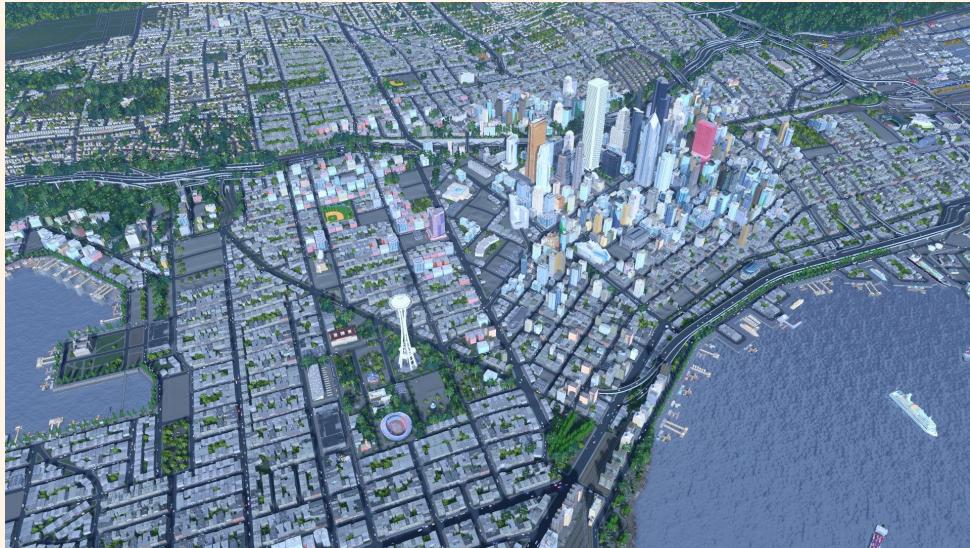
Ciudad Infestada,
Dark Souls



Performance test/benchmark

¿Para qué usarlo?

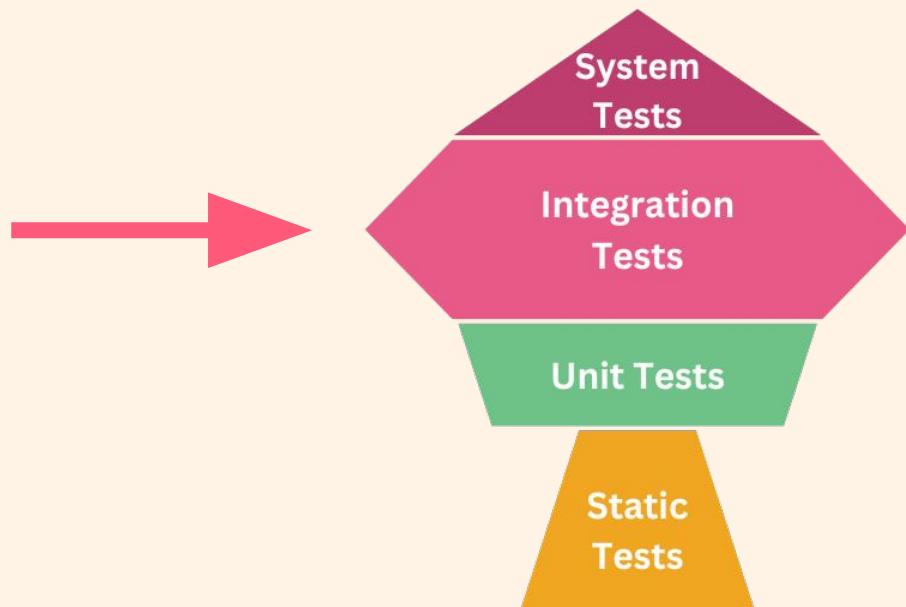
- Asegurar que objetos que aparecen a miles no se vuelven un problema



↑ Mesh de dentadura, Cities Skylines 2

← Cities Skylines 2 gameplay

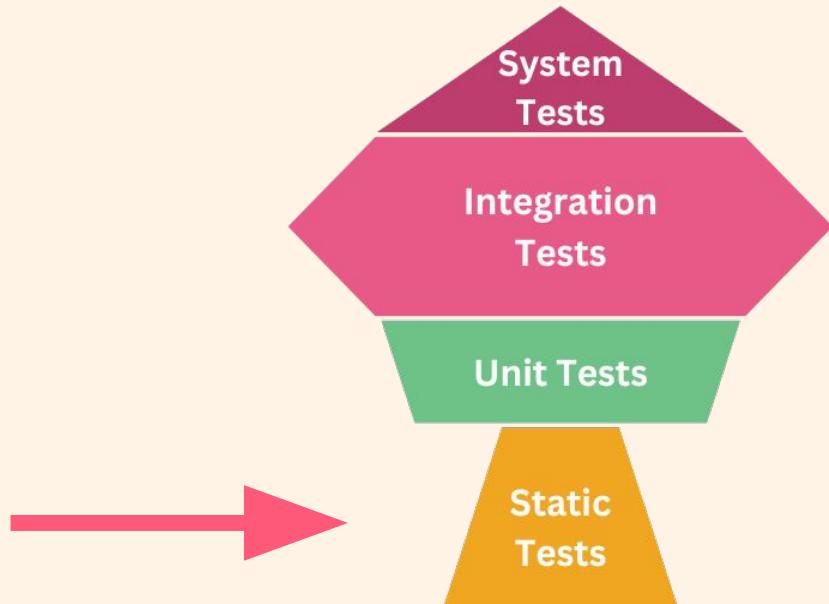
¡Más tipos de test!



Trophy testing model, Kent C. Dodds



Static analysis



Trophy testing model, Kent C. Dodds



Static analysis

- Sin ejecutar el juego, sólo leyendo los archivos del proyecto
- Dicen mucho y son **casi gratis**. Ejemplos:
 - Errores de compilación y linters
 - Script que revisa naming ficheros, configuración...

Icon2.svg: wrong file or folder name. Please, keep everything in lower case.

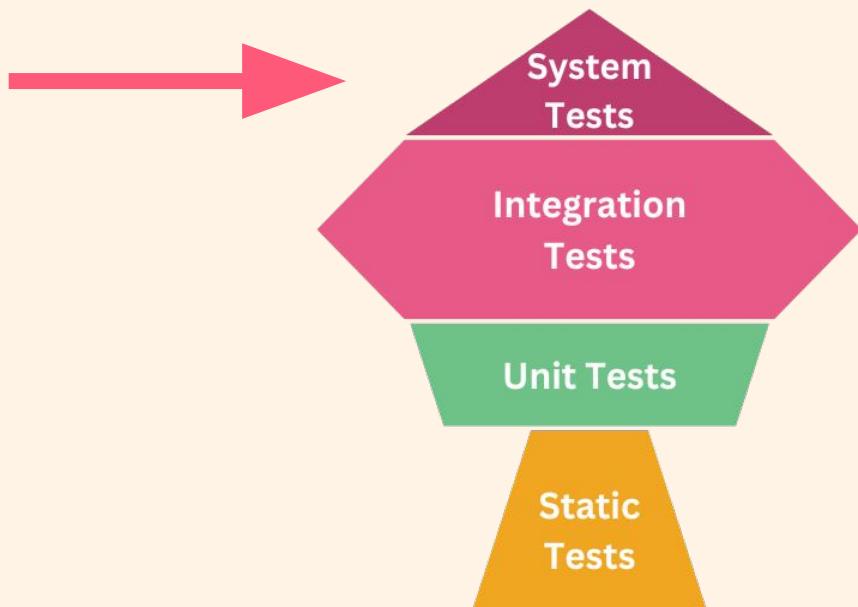
Icon2.svg.import: wrong file or folder name. Please, keep everything in lower case.

src/GameManager/game.tscn: wrong file or folder name. Please, keep everything in lower case.

src/input/WiiMoteInput.gd: wrong file or folder name. Please, keep everything in lower case.



Smoke test



Trophy testing model, Kent C. Dodds



Smoke test



Smoke test

Abre las escenas del juego para ver si tienen errores

```
74 - Testing res://src/core/game_menu/character_selection/character_selection.tscn...
75 - Testing res://src/core/game_menu/character_selection/player_selection_preview.tscn...
76 - Testing res://src/core/game_menu/character_selection/roster_character.tscn...
77 - Testing res://src/core/game_menu/credits/credits.tscn...
78 - Testing res://src/core/game_menu/customize_game_selection/game_mode_selection.tscn...
79 SCRIPT ERROR: Parse Error: Invalid argument for "new()" function: argument 4 should be "bool" but is "Resource".
80         at: GDScript::reload (res://src/core/game_menu/game_mode_selection/game_mode_selection.gd:22)
81 ERROR: Failed to load script "res://src/core/game_menu/game_mode_selection/game_mode_selection.gd" with error "Parse
error".
82         at: load (modules/gdscript/gdscript.cpp:3022)
83 - Testing res://src/core/game_menu/game_mode_selection/game_mode_selection.tscn...
```



Smoke test

 Tests Michiball APP 20:23 NUEVO

Smoke test execution found 34 errors  and 8 warnings .

Check the results in [this url](#).



Resumen

- Escribe tests, no demasiados, mayormente **integración**
- Tests para **problemas específicos**
 - Static analysis
 - Smoke test
 - Screenshot test
 - Performance test
 - ...
- Evita que algo malo ocurra **sin que lo sepas**



Parte 3

¿Cómo se automatiza?



Caso hipotético

¿Quién va a preparar un café de forma **más consistente**?

- Pablo, estudiante de uni, prepara 1L cada 2-3 meses cuando toca estudiar para los parciales, compra un café random del super...
- Máquina de cápsulas



Procesos manuales

- Un proceso **manual** es un proceso que **va a fallar**
 - Te interrumpen, te olvidas, te equivocas
 - Pantallazo azul, corte de luz, actualizaste un programa
 - ... y más
- **Evitar** los procesos manuales
 - O definir métodos de contención (ej. checklist)



¿Qué procesos podemos automatizar?

- ¡Lanzar **tests**!
- Generar las **builds** (Android/iOS, Web, Windows/Mac/Linux)
- Subir la build a **store** (itch.io, Steam, App Stores...)
- Tareas que sabe hacer sólo 1 persona
- Tareas que tengan que pasar cada X tiempo
 - (eg. limpiar un directorio, generar un changelog, generar documentación)
- Convertir imágenes/vídeos a un formato específico
- ...



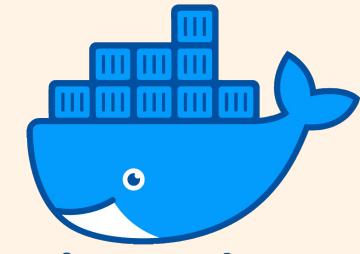
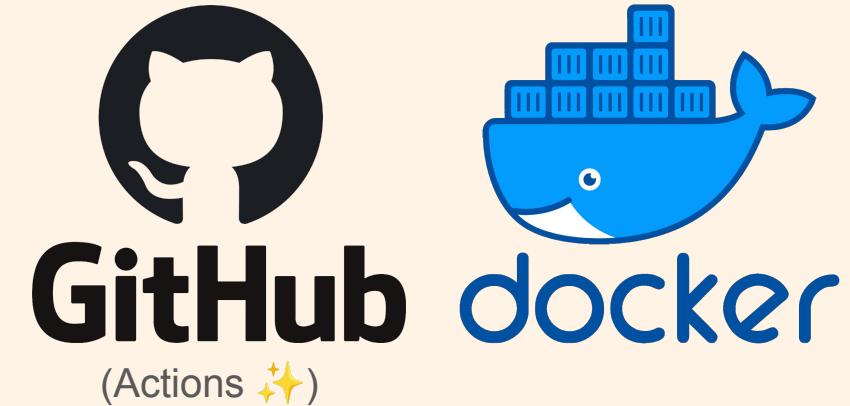
Integración continua (CI)

- Ejecutar scripts en **respuesta a**:
 - Subir un commit
 - Hacer merge de una pull request
 - Cada X tiempo
 - Pulsar un botón
- Automatización al **alcance** de todo el equipo
- Iterar más **rápido**, de forma más **consistente**



Integración continua (CI)

```
1 name: Build + Deploy
2 on: workflow_dispatch
3
4 env:
5   ITCHIO_USERNAME: nokorpo
6   ITCHIO_GAME: michiball
7   BUTLER_API_KEY: ${{ secrets.BUTLER_API_KEY }}
8   GODOT_VERSION: 4.4.1
9
10 jobs:
11   web:
12     name: Build and deploy to itch.io
13     runs-on: ubuntu-latest
14     container:
15       image: barichello/godot-ci:4.4.1
16     steps:
17       - name: Checkout
18         uses: actions/checkout@v2
19       - with:
20         lfs: true
21       - name: Setup
22         run: |
23           mkdir -v -p ~/.local/share/godot/export_templates
24           mv /root/.local/share/godot/templates/${GODOT_VERSION}.stable \
25             ~/.local/share/godot/export_templates/${GODOT_VERSION}.stable
26       - name: Web Build
27         run: |
28           mkdir build
29           godot --export-release --headless "Web" ./build/index.html
30       - name: Itch.io Deploy
31         run: |
32           ls ./build
33           butler push ./build ${ITCHIO_USERNAME}/${ITCHIO_GAME}:html5
```



Integración continua (CI)

```
1 name: Build + Deploy
2 on: workflow_dispatch
3
4 env:
5   - ITCHIO_USERNAME: nokorpo
6   - ITCHIO_GAME: michiball
7   - BUTLER_API_KEY: ${{ secrets.BUTLER_API_KEY }}
8   - GODOT_VERSION: 4.4.1
9
10 jobs:
11   web:
12     name: Build and deploy to itch.io
13     runs-on: ubuntu-latest
14     container:
15       image: barichello/godot-ci:4.4.1
16     steps:
17       - name: Checkout
18       - ...
19       - name: Setup
20       - ...
21       - name: Web Build
22       - ...
23       - name: Itch.io Deploy
24       - ...|
```

← Cuándo se ejecuta

← Variables:

Datos, nombres, passwords...

← Pasos a seguir



Integración continua (CI)

```
steps:  
  - name: Checkout  
    uses: actions/checkout@v2  
    with:  
      lfs: true  
  
  - name: Setup  
    run:  
      - mkdir -v -p ~/.local/share/godot/export_templates  
      - mv /root/.local/share/godot/templates/${GODOT_VERSION}.stable  
        ~/.local/share/godot/export_templates/${GODOT_VERSION}.stable  
  
  - name: Web Build  
    run:  
      - mkdir build  
      - godot --export-release --headless "Web" ./build/index.html  
  
  - name: Itch.io Deploy  
    run:  
      - ls ./build  
      - butler push ./build ${ITCHIO_USERNAME}/${ITCHIO_GAME}:html5
```

← Descarga repositorio

← Setup de Godot

← Generar build web

← Subir a itch.io



Integración continua (CI)

	Tiempo de ejecución	Storage
Github	1.000 min/mes	1 GB
CircleCI	2.000 min/mes	2 GB
Itch.io	–	30 GB

Consejo: salid del **monopolio** de Microsoft (GitHub)



Resumen

- Objetivo: encontrar problemas **lo antes posible**
- Testear e **iterar** procesos para conseguir **mejora continua**
- **Escribe tests**, no muchos, mayormente **integración**
 - Tests para **problemas específicos**
 - Evita que algo malo ocurra **sin que lo sepas**
- Evita error humano y cuellos de botella **automatizando**



¿Preguntas?



<https://nokorpo.com>

- Slides
- Más información
- Ejemplos
- Enlaces relevantes

