Taller de testing

Técnicas de programación concurrente

Testing en Rust

- attribute `#[cfg(test)]` para que se compile solo cuando corre cargo test
  - `#[cfg(not(test))]`
- attribute `#[test]` para marcar las funciones que son tests a correr
- las macros de `assert_eq!` son wrappers sobre panic()
- por defecto los tests se corren de forma concurrente

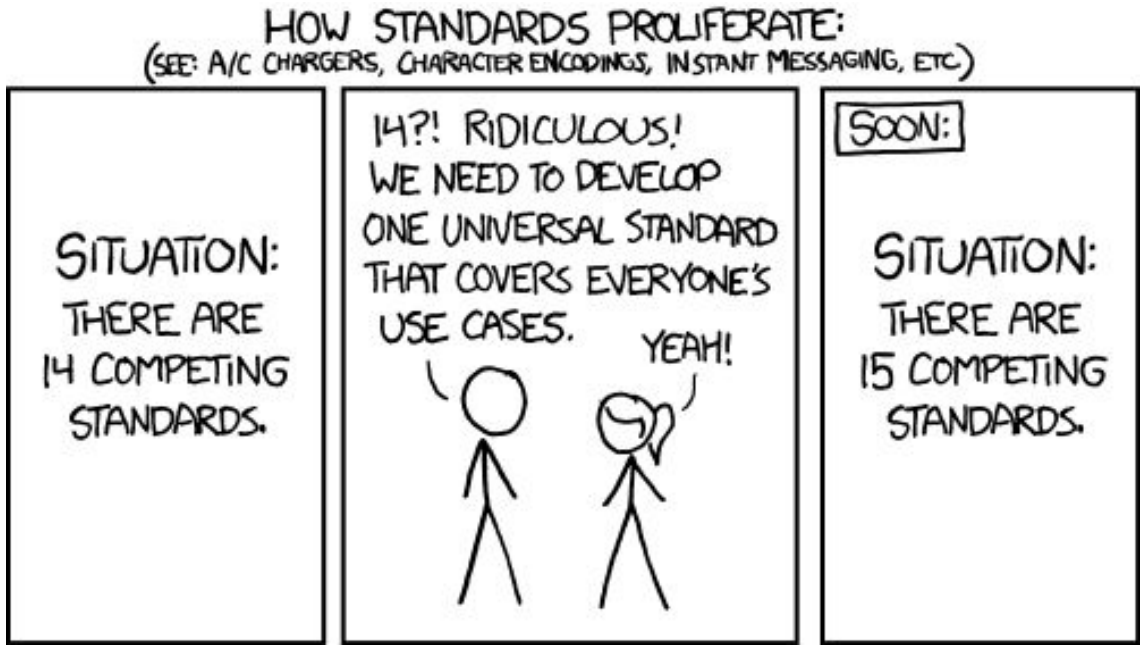
SOLID

- Single Responsibility
- Open/Closed
- Liskov Substitution
- Interface Segregation
- Dependency Inversion

El diseño determina al testeabilidad de un código

Ejemplo: juego de tirar una moneda. ver archivos solid.rs vs solid_refactor.rs

Crate mockall



https://asomers.github.io/mock_shootout/

Crate mockall

- ejemplos de mocks autogenerados de traits via `#[automock]`
- mocks de métodos estáticos
- uso de attribute `#[double]` para reemplazar una implementación en tiempo de test sin cambiar el resto del código
  - o manualmente via `#[cfg(not(test))]`

Race condition game

- Un grupo de threads duermen, cada uno por un tiempo aleatorio distinto
- El primero en despertar setea su id en un mutex
- El programa termina cuando todos despertaron
- El ganador se imprime por pantalla

Race condition game

- template methods para minimizar los cambios de código y conservar la esencia del algoritmo a testear
- reemplazo de sleeps no controlados por channels
- uso de "singletons" en rust via lazy_static para encontrar las instancias de los mocks a controlar

Loom

Loom is a tool for testing concurrent programs.

At a high level, it runs tests many times, permuting the possible concurrent executions of each test according to what constitutes valid executions under the C11 memory model. It then uses state reduction techniques to avoid combinatorial explosion of the number of possible executions.

Ejemplo de testeo de race condition game con loom cubriendo todos los posibles casos.

Actix

Testeamos el problema del banquero, versión de actores.

- Mocking de actores
- Tests async: esperando resultados sobre futures, channels o promesas (oneshot).
- Tests end-to-end: esperar por un mensaje final para detener el actor system
    - Con un timeout razonable