Hay muchos katas de programación muy conocidos que se utilizan para perfeccionar nuestras habilidades; sin embargo, hay una tendencia en estos katas de centrarse en un subconjunto muy pequeño de prácticas y a la vez se desarrollan sobre dominios sencillos que no son más complejos que un “juego de bolos” o un “juego de la vida”

Pero ¿qué pasa con todas las otras habilidades que un desarrollador de software necesita? Nuestro trabajo real es más complejo que desarrollar un único algoritmo, asimismo trabajamos en equipos y abordamos diferentes aspectos del software en paralelo.

Los *Code Katas* no ejercitan nuestras habilidades de comunicación o disciplinas como el diseño colaborativo, la integración continua y varias otras habilidades que se necesitan para entregar software que funcione al cliente.

El *Team Dojo* es un ejercicio de diseño y desarrollo colaborativo donde perfeccionaremos nuestras habilidades de trabajo en equipo. Los participantes formarán equipos para resolver en grupo un problema compartido: *Team Kata.* A cada equipo se le asignará un grupo de *historias de usuarios* y *pruebas de aceptación* que describen diferentes aspectos del problema. Cada equipo trabajará con cualquier estilo y lenguaje que desee, sin embargo se brindará ayuda especialmente para Java, C# y Ruby. Se tendrá 2 horas para pasar todas las pruebas de aceptación y crear un software que resuelva el problema. Todos están invitados a utilizar cualquier infraestructura para manejar el código y verificar los builds (Github, Hudson, etc). Los equipos no trabajaran de manera aislada, se brindará consejos e indicaciones si se solicita y así mantener las cosas siempre en movimiento. Estén preparados para utilizar su iniciativa!!!. Se otorgarán puntos a los equipos por cada prueba de aceptación que logren cumplir dentro del tiempo asignado.

No se trata solo de desarrolladores individuales creando buen código, sino también cómo los programadores individuales, trabajando juntos, pueden producir un gran software.

Se requiere que los asistentes tengan conocimiento intermedio de pruebas unitarias; no es obligatorio tener conocimiento de refactoring, integración continua y gestión de código, pero serán bastante útiles.

**Requisitos**

* IMPORTANTE: Tener conocimiento sobre pruebas unitarias.
* No es necesario tener conocimiento de ninguna herramienta de ATDD/BDD (Cucumber, Specflow, JBehave, etc).

# Descripción Corta

Team Dojo is an exercise in collaborative design and development. Participants will form into teams, each at their own table. Each team will be assigned a set of user stories and acceptance tests describing different aspects of the same problem. As a team, working in whatever style you wish and in whatever language you like (though help will be on hand for Java and .NET specifically), you will have 2 hours to create working software that solves the problem and passes all of the acceptance tests. The problem you'll be asked to solve will include a reapplication of a very famous network algorithm -[PageRank](http://en.wikipedia.org/wiki/PageRank) - to calculate the "kudos" of a person within a professional social network, as well as one of a range of shortest-path algorithms to calculate the number of degrees of seperation between people connected in the network. Having run 2-day team workshops on collaborative design, I promise that this session will test your abilities in communicating, collaborating and making decisions to the absolute limit. You're welcome to use any infrastructure you like to share and manage source code and corroborate builds (e.g., GitHub, Bamboo, Cloudbees etc). Although teams won't be spoonfed solutions, hints and pointers will be provided if requested to keep things moving. Be prepared to use your initiative! Points will be awarded to teams for every acceptance test they manage to pass within the time allotted.