



UNIVERSIDAD DE SONORA
DEPARTAMENTO DE INGENIERÍA
INDUSTRIAL
INGENIERÍA EN SISTEMAS DE
INFORMACIÓN



REPORTE FINAL DE PRÁCTICAS PROFESIONALES

PRÁCTICAS REALIZADAS EN NEARSOFT INC EN EL PROYECTO:

“NEARSOFT ACADEMY”

ASESOR DEL PROYECTO:

MARIO BARCELO VALENZUELA

ALUMNO:

OSCAR GERARDO SÁNCHEZ VALENZUELA

EXPEDIENTE:

208206154

CONTENIDO

1. INTRODUCCIÓN	4
2. OBJETIVO GENERAL.....	5
2.1 OBJETIVOS ESPECÍFICOS	6
3. PLANTEAMIENTO DEL PROBLEMA	7
4. PROPUESTA DE SOLUCIÓN.....	8
4.1 ALCANCE Y LIMITACIONES DEL PROYECTO.....	9
4.2 JUSTIFICACIÓN.....	10
5. METODOLOGÍA APLICADA	11
6. CONCLUSIONES Y RECOMENDACIONES	12
8. ANEXOS	13
Academy Outline (the Ninja Roadmap).....	13
White belt (1 month) - Half time, four weeks.....	13
Green - Brown belt (2 months).....	14
Red Belt (1 month).....	15
Black Belt.....	16
Interns regulation.....	17
Table of Contents.....	23
Quality Assurance Test Plan.....	23
1. Document Overview	23
1.1. Doc.....	23
Document Organization.....	23
1.2. Document Review Team.....	23
2. Test Plan Strategy.....	23
2.1. Test Plan High Level Description	23
2.2. Test Plan Scope.....	23
2.3. Test Plan Scope Exclusions.....	23
2.3.1. Test Plan Schedule Dependencies	23
2.4. Testing Areas Common to All Projects.....	23
2.4.1. Multiple Users Testing	23
2.4.2. Database References	23

2.4.3.	Dependencies.....	23
2.4.4.	Test Data	23
2.4.6.	Automated Tests.....	23
2.5.	Testing Time Required.....	23
2.5.1.	Estimated time required: Best estimate	23
2.5.2.	Actual time required: Actual number of person-hours.....	23
2.5.3.	Discrepancies between Actual time and Scheduled time	23
3.	Test Cases – Detailed Descriptions of Tests □.....	23
Document Overview		23
1.	Document Organization	23
2.	Document Creation Team.....	23
1.	Core Team	23
Test Plan Strategy		24
3.	Test Plan High Level Description.....	24
4.	Test Plan Scope	24
5.	Test Plan Scope Exclusions.....	24
1.	Test Plan Schedule Dependencies	24
6.	Testing Areas Common to All Projects.....	24
1.	Multiple Users Testing	24
2.	Database References	25
3.	Dependencies.....	25
4.	Test Data	25
5.	Automated Tests.....	25
7.	Testing Time Required	25
1.	Estimated time required: Best estimate	25
2.	Actual time required: Actual number of person-hours.	25
Test Cases – Detailed Descriptions of Tests.....		25
Test Case UWGEN01		26
Test Case UWGEN02		26
9.	BIBLIOGRAFÍA	29

1. INTRODUCCIÓN

En el siguiente documento se van a describir las actividades realizadas en la empresa de desarrollo de software Nearsoft, específicamente en el proyecto llamado Nearsoft Academy (1), que fue creado para ayudar a los practicantes a educarse en el ámbito laboral y consolidar sus conocimientos adquiridos a lo largo de la carrera.

Dentro de Nearsoft Academy, el interno llevar a cabo distintas tareas. En este proyecto se van a describir cada una de las fases de la Nearsoft Academy y la ayuda que aporté en la creación de la misma.

Nearsoft buscando el desarrollo y la innovación empresarial en Hermosillo, apoya iniciativas para la creación de nuevas empresas, y así mejorar la economía y la generación de empleos en la localidad.

Además de ayudar con Nearsoft Academy, se colaboró en el aseguramiento de la calidad en el proyecto Uwisio.

2. OBJETIVO GENERAL

La Nearsoft Academy quiere dar algo de vuelta a la comunidad del software de código abierto, entrenar el talento y proveer y compartir conocimientos en un entorno práctico.

El objetivo de ayudar en el aseguramiento de calidad en Uwisio era, como su nombre lo dice, verificar que el producto final fuera satisfactorio para el usuario.

2.1 OBJETIVOS ESPECÍFICOS

El objetivo del proyecto para apoyar a la Nearsoft Academy fue ideado principalmente por Isaác López, programador de Nearsoft, para ayudar a traer nuevos reclutamientos para trabajar en Nearsoft. Este proyecto consistía en hacer una guía tematizada en un camino ninja, con la cuál los internos aprenderían desde las raíces de la programación hasta llegar a hacer un proyecto por sí solos.

El objetivo de ayudar a Uwisio era muy parecido al general. Elaborar casos de uso, casos de prueba y ejecutarlos con el fin de encontrar errores en el sitio web de Uwisio.

3. PLANTEAMIENTO DEL PROBLEMA

Muchas personas tienen lo que se necesita para entrar a Nearsoft pero de todos modos les faltan habilidades técnicas o un poco de conocimiento para aplicar sus habilidades. Para eso se creó Nearsoft Academy.

Por la parte de Uwisio, se analizó el sitio web y se encontraron muchos errores que, en caso de arreglarse, darían una vista mucho más profesional al mismo.

4. PROPUESTA DE SOLUCIÓN

Para la Nearsoft Academy se creó un plan de acción con los objetivos de entrenar el talento, y enseñar mediante la práctica, además de ayudar a candidatos a ganarse una buena reputación por ellos mismos. De esta manera Nearsoft ganaba excelentes y competentes nuevos reclutamientos, además de un foco de conocimiento.

Para Uwisio se propuso crear un plan de pruebas, conteniendo las diferentes áreas del sitio web y diferentes casos de pruebas aplicados a las áreas contenidas. Estos casos de prueba serían ejecutados con el fin de encontrar desperfectos que pudieran dañar la experiencia del usuario final.

4.1 ALCANCE Y LIMITACIONES DEL PROYECTO

La Nearsoft Academy fue planeada para poderse extender fuera del estado, y, además, que no tuviera problemas para migrarse de un lenguaje de programación a otro o hasta de una metodología a otra. Su limitación era que el proyecto depende enteramente de la responsabilidad del practicante para prosperar correctamente.

El alcance actual de Uwisio es la extensión del sitio web por todo el estado de Sonora, y sus limitaciones son claras al cubrir sólo partidos de deportes locales/estatales.

4.2 JUSTIFICACIÓN

El apoyo a los proyectos de Nearsoft Academy y Uwisio fue realizado con el fin de aprender nuevas técnicas, apoyar a la creación de una guía y mejorar la experiencia del usuario final en ambos proyectos.

5. METODOLOGÍA APLICADA

Se utilizó programación externa (XP) para colaborar con módulos controlados por tiempos determinados, de una manera más ordenada y eficiente. No se utilizaron lenguajes de programación en específico.

6. CONCLUSIONES Y RECOMENDACIONES

En mi opinión, más personas deberían hacer lo posible por hacer las prácticas en un lugar en el que en realidad quisieran trabajar, ya que, por ejemplo, Nearsoft es una empresa completamente libre, responsable y con un foco de conocimiento increíblemente superior a la media. Con esto me refiero a que con el simple hecho de hacer las prácticas profesionales ahí, el estudiante podrá adquirir conocimientos que le servirán en cualquier otra empresa a la que quiera ingresar.

Mi recomendación personal es que no se den por vencidos porque no pasaron el examen una vez. El que persevera alcanza, y en este caso lo que se alcanza da muchas ventajas laborales en el futuro.

8. ANEXOS

The ninja roadmap (guía para internos de Nearsoft)

Academy Outline (the Ninja Roadmap)

Duration: 6 months

White belt (1 month) - Half time, four weeks

Goals:

- Reset expectations
- Develop a wide tech vision

Topics

- Pretotypes
- Lean-Agile
- Innovation (myths)
- Nearsoft philosophy

Must have

[Pragmatic programmer](#)

Getting Things Done by David Allen

Books to Read (at least one)

- [Tribal leadership](#) (?)
- [Start with Why](#)
- [Good to great](#) by Jim Collins

[The Design of Everyday Things](#)

[Don't Make Think](#)

[Effective Programming](#) by [Jeff Atwood](#).

Deliverables

- Three (two) tested pretotypes (different ideas)
- Book Summary or Review
- 4 blog post about talks (infoq, google io)

Green - Brown belt (2 months)

Goals:

- Build something useful
- Learn through practice
- Work as a team

Topics

- Rapid development Lang - framework (Grails, Ruby, Play, CakePHP)
- Build a project from scratch *and finish it*

Must have

- Continuous integration(**source control**, release management, automated tests).
- **OWASP**.
- **Scrum or Kanban** practices.
- **Logging** & Analytics.

Nice to have

- File uploads. / *Emailing or IM with a XMPP*
- Caching
- Regular Expressions

Books to Read

- Code Complete
- Pragmatic Project Automation
- *OTHER PROPOSALS*
 - [The Art of Unit Testing](#)
 - The Mythical man-month

Books for Specific Technologies

Mastering Regular Expressions

- [JavaScript: The Good Parts](#)
- [C# In Depth](#)
- [Effective Java](#)

Deliverables

- Feature Complete App (Green belt)
- Running App with real users (Brown belt)
- 8 blog post about talks (infoq, google io)
- 1 Book resumes

Red Belt (1 month)

Goals:

- Open source culture
- Learn to read and understand other's code
- Adding something worthy to the resume

Topics

- Open source contribution
- Written communication
- One hour daily pseudo-quiet pair with a Nearsoft developer

Code to Read

- Chromium
- Hibernate
- Ant
- Test NG
- JUnit
- Google guava
- Tomcat
- Doom 3
- Node.js
- Mockito

Books

- [The Cathedral and the Bazaar](#)
- [97 Things Every programmer should know](#)

Deliverables

- 4 blog post about OS projects (community, source code highlights, what do I learn)
- 1 Book resume
- 1 Accepted contribution
- 300? StackOverflow Reputation

Black Belt

Goals

- Get a Job

Topics

- Master the job interview
- Pimp my resume

Books to Read

- [Masterminds of Programming](#)
- [Coders at Work](#)
- [97 Things every software architect should know](#)
- [Masters of DOOM](#)
- [Hackers: Heroes of the Computer Revolution](#)

Deliverables:

- Passed client interview
- 4 simple & useful webcasts (e.g. <http://railscasts.com>)

Interns regulations (reglas para practicantes recién ingresados en

Nearsoft)

Interns regulation

- No hay un horario específico de llegada ni de salida mientras se haga el trabajo a tiempo y correctamente.
- Las daily meetings deben ser puntuales y se deben exponer los obstáculos al mentor para que este haga lo posible por resolverlos o buscar a la persona adecuada para ayudar.
- Si debes faltar por causas de fuerza mayor, haz lo posible por avisar con tiempo para que no haya ningún problema.
- La entrega de las tareas deberá hacerse dentro del tiempo acordado por el mentor y el interno.
- No deberás cerrarte a tus conocimientos y escucharás a tu mentor, sus sugerencias y retroalimentación. ([Be water, my friend](#))
- Deberás prestar atención y ser responsable.
- Deberás actuar 'Like a boss' para salir adelante.
- Deberás mostrar tus mejores habilidades para hacer todo lo que se te pida, incluyendo imaginación y originalidad.
- Y sobre todo, ¡DIVIÉRTETE MIENTRAS APRENDES!

The ninja roadmap – Interns guide (Guía para internos de la Nearsoft Academy)

The Ninja Roadmap: Interns Guide

Hello, ninja apprentice. You've just entered to the guide to success. Through this roadmap, you will need to:

- Pay attention and be responsible.
- Act like a boss and like an adult.
- Show your best skills to do everything asked.
- [Be water, my friend](#)

The objective of this roadmap is to teach you the deepest secrets in order to reach a better level of knowledge, you need to be patient and you will need to empty your mind, forget everything and listen carefully to your mentor. That is the key in everything: learn as much as you can, because every new concept you learn is the bigger your knowledge database will be.

You will travel through different paths, some easier than the others but, in the end, the experiences gained in your roadmap (with the help of your sensei, your mentor) will be the needed to get a job a Nearsoft.

Each month you will be reading a different book based on the topic in that belt. After you read that book, you will need to write a short review to let the people at Nearsoft know what the book talks about. In addition, you will be watching and listening to different videos and making a summary about it. Also, you will be uploading your summaries and reviews to the Nearsoft Academy [website](#), so the world can learn from your learning (yes, it sounded kind of redundant).

I will start introducing you belt by belt, in order to give you some useful information to support your tasks.

White belt (1 month) – 4 hours daily, four weeks

This is the first month, the first challenge and the easiest one. Here your expectations and preferred programming languages will be resetted. As Bruce Lee said once, "[Empty your mind](#)". This month is your white belt training. You're still a chick who just came out from the egg.

Topics in this month:

- Prototypes
 - [Prototyping general information](#)
 - [Prototyping e-book 'Prototype it'](#)
 - [Prototyping book 'The lean startup'](#)
 - [How to take user feedback](#)
 - [Vision or hallucination?](#)
- Lean-Agile
 - [Agile at Google](#)
 - [Scrum](#)
 - [Agile testing](#)
- Innovation (myths)
 - [Innovation at Google](#)
 - [The myths of innovation](#)
 - [The process of innovation](#)
- Nearsoft philosophy
 - [El paradigma Balboa: soy estúpido pero aguanto los golpes](#)

Week 1: thinking three ideas and their three “pretotypes”

This is the first week, welcome to your first challenge. Here you will learn some concepts about how to make your ideas come true by testing them, thinking them and “pretotyping” them.

In this step, you will think about three useful ideas and, while you create them, you will read articles and watch videos about that topic. You will have to deliver three ideas with one “[pretotype](#)” each.

Week 2, 3 and 4: testing your pretotypes for each idea

Each week, you will be testing your idea by working on your pretotype, extracting information, analyzing data, making new decisions. These ideas will be tested, evaluated and you will choose one of them at the end. To test them, you will need to study your environment, your community and your market and see if it could really work in a future, see if people would like it.

Books you can read this month:

- Important:
 - Pragmatic Programmer
 - Getting things done
- Secondary:
 - Start with why
 - Good to great
 - The design of everyday things
 - Don't make me think

Deliverables at the end of the month

- Three tested prototypes from your three different ideas and one final decision.
- Book summary.
- 4 blog post about talks.

Green/Brown belt (2 months) – 4 hours daily, eight weeks

By this time you have probably finished your first month on Nearsoft. You already started to know everyone and hopefully you already feel comfortable too. Also, you already picked one idea from the three you thought about.

These are the second and third belt, one of the most challenging steps of this roadmap. Basically what you will do is an application using the knowledge acquired in the first month and making your idea bigger, stronger and better.

In the first month you will work hard until your app is ready, complete and fully working with every feature planned and every need asked. In the second month you will need to test and run your app with real users to see if it works as planned and to see if it causes the expected reactions in the users.

Your most important goals are:

- **Build something useful:** you need to create an application not only with the purpose of passing this month and nothing else. You need to create something useful for the world, something new, and something that people will keep using once they used it for the first time. It probably exists already, but you need to think about something you could improve in it.

- **Learn through practice:** the best way to learn something is using the kinesthetic learning, which means that you need to interact with the knowledge in order to fully learn it.
- **Work as a team:** along with your mentor, you will finish your application by the end of these two months.

Topics in this two months:

- Rapid development languages / frameworks:
 - o [Grails](#)
 - o [Ruby](#)
 - o [Play](#)
 - o [CakePHP](#)
- Build a project from scratch and finish it

What your project will need to have:

- o [Source control](#)
- o [OWASP.](#)
- o [Scrum practices](#)
- o [Logging.](#)

Books you can read these two months:

- Important:
 - o Code Complete
 - o Pragmatic Project Automation
 - o [The Art of Unit Testing](#)
 - o The Mythical man-month
- Books for Specific Technologies
 - o Mastering Regular Expressions
 - o [JavaScript: The Good Parts](#)
 - o [C# In Depth](#)
 - o [Effective Java](#)

Deliverables at the end of the month

- Feature Complete App (Green belt)

- Running App with real users (Brown belt)
- Book summary.
- 8 blog post about talks.

Quality assurances test plan at Uwisio (Plan de aseguramiento de calidad y pruebas en Uwisio)

Quality Assurance Test Plan at Uwisio

Table of Contents²

Quality Assurance Test Plan

1. Document Overview

1.1. Doc

Document Organization

1.2. Document Review Team

2. Test Plan Strategy

2.1. Test Plan High Level Description

2.2. Test Plan Scope

2.3. Test Plan Scope Exclusions

2.3.1. Test Plan Schedule Dependencies

2.4. Testing Areas Common to All Projects

2.4.1. Multiple Users Testing

2.4.2. Database References

2.4.3. Dependencies

2.4.4. Test Data

2.4.6. Automated Tests

2.5. Testing Time Required

2.5.1. Estimated time required: Best estimate

2.5.2. Actual time required: Actual number of person-hours.

2.5.3. Discrepancies between Actual time and Scheduled time

3. Test Cases – Detailed Descriptions of Tests

Document Overview

1. Document Organization

This document follows the standard Web Team specification format with some modifications as needed for clarification.

2. Document Creation Team

1. Core Team

- [Javier Rojas](#)
- [Gerardo Sánchez](#)

Test Plan Strategy

In this section we are going to explain every part of the test plan, describing the plan, its scope and the testing areas of the project.

3. Test Plan High Level Description

This plan's strategy is to test every single area from the website in order to maintain the actual quality. The idea is to view the site from another perspective besides the developer team, finding common mistakes while acting like a normal user.

The quality is going to be verified with manual test cases and exploratory testing in order to give the team new suggestions to improve Uwisio.

4. Test Plan Scope

This plan includes the following areas of the project to test:

- [General](#)

- [Calendar](#)

- [News](#)

- [Restaurants](#)

- [Special offers](#)

- [User](#)

- [Spreading the word \(Social Networks\)](#)

5. Test Plan Scope Exclusions

1. Test Plan Schedule Dependencies

- Facebook Javascript API.

- Picasa API.

- Twitter API.

- Twitter Widget.

6. Testing Areas Common to All Projects

1. Multiple Users Testing

The multi-user tests are going to be done in order to find out if there is any danger while two or more users are looking at the website or logged in the same account at the same time. The approach of this is to optimize the way that users get access to the site and to reduce the chances of vulnerability.

2. Database References

There is going to be database tests, looking for weaknesses in it while inserting lots of data at the same time and watching if there is any register missing at the end. The plan is to make the database even stronger against common issues, like data redundancy.

3. Dependencies

There are some [dependencies](#) which may impact in the execution or the performance of Uwisio, but can't be fully tested because they work apart from the site. The best that can be done is to improve the use of them or modifying the total dependency of them.

4. Test Data

Real data and test data will be used. Test data will be created manually by the QA team (test users and restaurants). This is to identify how the system behaves with different data.

5. Automated Tests

The plan is to automate some of the most common tests, like checking the database or edit an user. The end of this is to focus on new tests while the common ones are being done automatically.

7. Testing Time Required

1. Estimated time required: Best estimate
2. Actual time required: Actual number of person-hours.

Test Cases – Detailed Descriptions of Tests

General

Link and code	Description of the test case
UWGEN01	x Verify that the text translates correctly when changing the language
UWGEN02	A description is shown when hovering over an image in home.

Test Case UWGEN01

Purpose:

1. Verify that the text translates correctly when changing the language of the site.

Test Data:

Action	Input	Expected Output
Launch the application	http://uwisio.com/	The website loads the home page.
Verify that English and Spanish languages are available.		Verify that the English and Spanish options appear at the bottom of the website.
Select a language	Select "English" option.	The website reloads and the text is translated to English.
Return to default	Select "Español" in the combo box.	The website reloads and the text is translated to Spanish.

Test Case UWGEN02

Purpose:

1. A description is shown when hovering over an image in home.
2. The description corresponds with the image.

Test Data:

Action	Input	Expected Output
Launch the application	http://uwisio.com/	The website loads the home page.
Select an option from the news menu.	Select the first option from the news menu.	Verify that an image shown on the left side of the menu corresponds with the option selected.
Hover the cursor		Verify that a description to the image appears

over the image.		and corresponds with the option selected in the menu.
-----------------	--	---

Calendar

Link and code	Description of the test case
UWCAL01	Observe that the page change in the calendar works correctly
UWCAL02	Verify that the teams' photos matches with the current team
UWCAL03	The calendar only shows the current leagues of the season.
UWCAL04	Check that all the teams have a photo by their side in the calendar
UWCAL05	Test that all the calendar functionalities work on the 5 most important browsers
UWCAL06	The icon shown in every tab in the calendar corresponds to each sport.
UWCAL07	Check that all the tabs in the calendar's sports must have their own pictures according to the sport
UWCAL08	Verify that the matches' dates are correct
UWCAL09	Verify that the matches' schedules are correct

News

Link and code	Description of the test case
UWNEW01	

Restaurants

Link and code	Description of the test case
UWRES01	The restaurant's location is shown accurately in the map.

Special offers

Link and code	Description of the test case
UWSPE01	

User

Link and code	Description of the test case
UWUS01	The user can log-in to the site correctly

Spreading the word

Link and code	Description of the test case
UWSPR01	

9. BIBLIOGRAFÍA

1. <http://academy.nearsoft.com/>
2. <http://uwisio.com/>