



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN

Departamento de Sistemas Informáticos y Computación

Diseño de un modelo y construcción de una herramienta de apoyo para evaluar prácticas ágiles aplicables en equipos de trabajo

Tesis de fin de Máster

Máster Universitario en Ingeniería del Software, Métodos Formales y Sistemas de
Información

Autor: Mario Pérez Alarcón

Director: Patricio Letelier Torres

Julio de 2014

Resumen

Trabajo de fin de máster en el que se ha construido una herramienta web que basada en un modelo facilita la implantación de prácticas ágiles en equipos de trabajo.

El modelo establece relaciones entre los elementos que permiten decidir qué prácticas ágiles convendría aplicar en un equipo. Estos elementos son: la contribución de prácticas a objetivos de mejora, el esfuerzo de preparación de cada práctica, nivel de aplicación actual de cada práctica, otras prácticas asociadas y posibles desafíos de las prácticas.

La herramienta desarrollada permite evaluar de forma conjunta los elementos antes señalados. Para su implementación se han utilizado los frameworks de código libre Ruby on Rails y Bootstrap.

Palabras clave: Métodos ágiles, Prácticas ágiles, Implantación de agilismo.

Tabla de contenidos

Figuras	8
1. Introducción.....	11
2. Objetivos.....	13
3. Implantación de un método ágil	15
Métodos Ágiles.....	15
Ciclo de Vida Ágil.....	15
Implantación	16
4. Estado del Arte	19
Comparative Agile Assessment (CA).....	19
Agile Maturity Model (AMM)	21
Enterprise Agility Maturity Matrix	22
Conclusión.....	22
5. Agile Roadmap+.....	23
Punto de partida. Agile Roadmap.....	23
Agile Roadmap+.....	23
Modelo.....	24
Características Principales sitio Agile Roadmap+	26
Proceso de Desarrollo.....	31
6. Ecosistema tecnológico usado en Agile Roadmap+.....	33
 Ruby on Rails.....	33
Ruby	33
Rails.....	37
Instalación de Rails.....	37
Estructura de carpetas	38
Base de Datos	40
Comandos útiles Rails/Rake.....	41
 Bootstrap.....	43
Móviles	43
Tablets	44

Desktops	45
 Git.....	45
Ramas	46
 GitHub.....	46
Branching Model	48
 Heroku	49
	50
Dokku	51
Herramientas y Recursos Utilizados.....	52
 Sublime Text 3	52
 RubyMine	52
	52
 Inspector de Google Chrome	52
Git Shell.....	53
 SQLite Database Browser	53
Putty.....	53
	53
	53
	53
7. Guía de uso	55
8. Estadísticas Agile Roadmap+	59
	59

Sesiones	59
País	60
Navegador.....	60
Sistema Operativo Móvil.....	61
Flujo del comportamiento.....	61
Sitio Agile Roadmap+	62
Usuarios	62
Posición media de cada práctica	64
Nivel medio de aplicación de cada práctica	65
9. Conclusiones y Trabajo Futuro	67
Referencias	69

Figuras

Figura 1. Beneficios agilismo, State of Agile Survey, 2014	12
Figura 2. Obstáculos adopción agilismo, State of Agile Survey, 2014	12
Figura 3. Evolución hacia el agilismo [1]	17
Figura 4. Ejemplo Comparative Agility	20
Figura 5. CA - Cuándo ocurre la planificación	21
Figura 6. Agile Maturity Model	22
Figura 7. Modelo AgileRoadmap	24
Figura 8. Agrupación por método ágil	25
Figura 9. Objetivos específicos de mejora.....	26
Figura 10. Catálogo Prácticas.....	27
Figura 11. Detalle Práctica	27
Figura 12. Inicio del dragdrop	28
Figura 13. Fin del drag&drop	28
Figura 14. Ejemplo de evaluación práctica	28
Figura 15. Ver Prácticas Relacionadas.....	29
Figura 16. Desafío Práctica	30
Figura 17. Objetivos de mejora	30
Figura 18. Contribuciones de las prácticas a un objetivo	31
Figura 19. Unidades de Trabajo	31
Figura 20. Flujo de Trabajo	32
Figura 21. ERB - Haml.....	34
Figura 22. Barra progreso Youtube	36
Figura 23. Chartkick.....	37
Figura 24. Agile Roadmap+ en móviles.....	44
Figura 25. Agile Roadmap+ en tablets	44
Figura 26. Agile Roadmap+ en pc.....	45
Figura 27. Git ramas	46
Figura 28. Github commits.....	47
Figura 29. Github frecuencia commits	47
Figura 30. Git Branching Model.....	48
Figura 31. Heroku Dynos	50
Figura 32. Vídeo ayuda Agile Roadmap+	55
Figura 33. Sesiones Analytics	59
Figura 34. Analytics -- Sesión/País	60
Figura 35. Navegadores Analytics.....	60
Figura 36. SO móvil Analytics	61
Figura 37. Flujo del comportamiento Analytics.....	61
Figura 38. Ámbito de trabajo.....	62
Figura 39. Sector de empresa	63
Figura 40. Número de trabajadores	63
Figura 41. Países con usuarios registrados	64

Figura 42. Posición Prácticas Ágiles	64
Figura 43. Nivel medio de aplicación de cada práctica.....	65

1. Introducción

“Los métodos ágiles para la gestión de proyectos y equipos de trabajo acaparan la atención de la industria, sin embargo, existe resistencia y desconfianza para su implantación, especialmente en entornos acostumbrados a trabajar de una forma más tradicional. Un obstáculo importante para que las empresas se embarquen en iniciativas de implantación de métodos ágiles es la falta de un cuerpo de conocimiento consensuado y de guías para llevar a cabo la implantación del agilismo, considerando posiblemente la convivencia con ciertas prácticas tradicionales. Otro desafío, y para no cometer fallos de otras estrategias de implantación de mejoras de proceso, es que la aplicación de prácticas debe ser configurable a las necesidades específicas del contexto de trabajo del equipo, es decir, cada equipo debería tener un proceso ajustado a sus necesidades, determinadas también por los productos o servicios que le han sido encargados.” [1].

Es vital el diseño de un modelo para asistir a las empresas en dicha implantación. El esfuerzo requerido para realizar la transición al agilismo suele ser subestimado. Por ejemplo, en [2] se mencionan los principales problemas que surgen al implantar Scrum:

- **Un cambio satisfactorio no es enteramente *top-down* o *bottom-up*:** el cambio deberá realizarse teniendo en cuenta ambas perspectivas.
- **El estado final es impredecible:** es muy difícil y en ocasiones imposible predecir exactamente como responderán las personas a los muchos pequeños cambios que serán necesarios para implantar el agilismo.
- **Scrum es pervasivo:** es pervasivo en el sentido de que invade todo lo que hace un desarrollador a lo largo del día en el trabajo.
- **Scrum es dramáticamente diferente:** es muy diferente en comparación a lo que se había hecho hasta hoy.
- **El cambio viene cada vez más rápido:** muchos cambios en un corto periodo de tiempo puede desorientar y estresar a los empleados.
- **Las buenas prácticas son peligrosas:** las buenas prácticas pueden relajarnos y parar el proceso de continua mejora que es esencial en Scrum.

Por otra parte, los beneficios obtenidos una vez lograda una correcta transición pueden ser muy importantes. Entre las razones más mencionadas para realizar la transición al agilismo se encuentran:

- **Mayor productividad y menor coste.**
- **Mejor compromiso y satisfacción del empleado al realizar su trabajo.**
- **Mayor calidad.**
- **Mejora en la satisfacción del cliente.**
- **Lo que hasta ahora hemos estado haciendo ya no funciona.**

En la encuesta sobre el estado actual en agilismo (ver Figura 1) se encuentran ilustradas las razones más importantes para realizar la transición al agilismo.

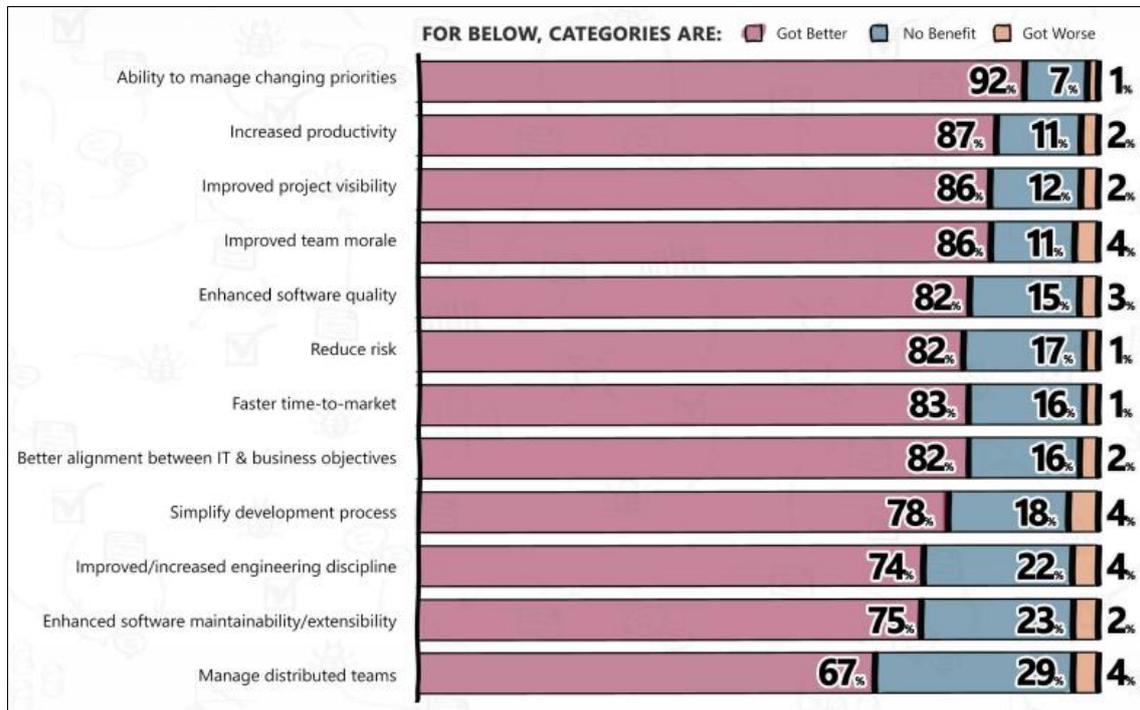


Figura 1. Beneficios agilismo, State of Agile Survey, 2014

En esta encuesta también se recogen los principales obstáculos (ver Figura 2), siendo el principal de ellos la incapacidad de cambiar la cultura organizacional de la empresa.

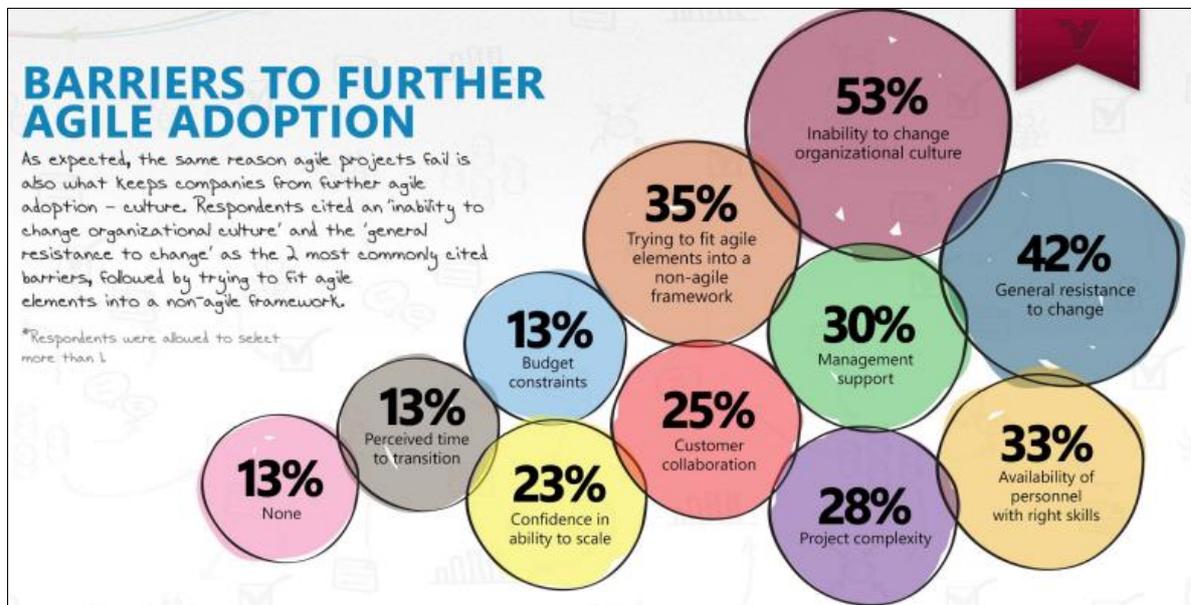


Figura 2. Obstáculos adopción agilismo, State of Agile Survey, 2014

2. Objetivos

El objetivo principal de este trabajo de tesis es aplicar un modelo para evaluación de prácticas ágiles que se implementarán en un contexto específico de equipo y producto o servicio. El modelo debe abordar factores tales como la contribución de cada práctica a los objetivos del negocio, la importancia de cada objetivo, el margen de mejora que puede ofrecer una práctica en el contexto específico, los desafíos de cada práctica, las relaciones entre prácticas, etc. Se construirá una herramienta software que apoyará en la evaluación de prácticas con el propósito de ofrecer a un interesado realizar una autoevaluación basada en el modelo. La herramienta considerará aspectos específicos de usabilidad para facilitar la gestión de todos los factores incluidos en el modelo.

El software a construir consistirá de una plataforma web usando tecnologías punteras como son: Ruby on Rails, Bootstrap, HTML5, CSS3, JQuery, Git. En [3] se indican las ventajas de construir el software en forma de una aplicación web:

- Ubicuidad y conveniencia de usar un navegador web como cliente.
- Compatibilidad multiplataforma.
- Posibilidad de actualizar y mantener la aplicación sin tener que distribuir e instalar el software en multitud de clientes.
- Reducción en los costes asociados a las tecnologías de información.

3. Implantación de un método ágil

Métodos Ágiles

Los métodos ágiles tienen su origen en el año 2001, momento en el que se publicó el Manifiesto Ágil [4] (en inglés, *Agile Manifesto*). En éste, se detallaban nuevas formas de desarrollar el software de manera más eficiente, en el manifiesto se valoran principalmente:

- Individuos e interacciones por encima de procesos y herramientas.
- Software en funcionamiento por encima de documentación comprensiva.
- Colaboración con los clientes por encima de negociación de requisitos.
- Respuesta al cambio por encima del seguimiento de un plan.

Mientras que existe un valor en los ítems de la parte derecha, se valoran más los ítems de la parte izquierda.

Los métodos ágiles causaron polémica cuando aparecieron: "... yet another attempt to undermine the discipline of software engineering... nothing more than an attempt to legitimize hacker behavior." [5]. Pero en la actualidad son ampliamente aceptados: en un estudio de 66 proyectos se constató que la mayoría de ellos usaban metodologías ágiles, incluso para equipos distribuidos. [6] Además, en la última encuesta realizada por *Versionone* [7] se obtuvo que el 88% de las empresas participantes (un total de 3501) aplicaban métodos ágiles para el desarrollo de sus productos.

Ciclo de Vida Ágil

En el ciclo de vida de un desarrollo ágil se destaca:

- Aceptación de los cambios como un hecho de la vida. Mejora continua.
- Los desarrolladores refinan el trabajo de un prototipo hasta que el cliente se quede satisfecho. El cliente proporciona *feedback* en cada iteración (de no más de 1 mes de duración).
- Se enfatiza el desarrollo dirigido por pruebas o TDD (en inglés, *Test-Driven-Development*) para reducir fallos, historias de usuario escritas para validar así los requerimientos del usuario, y la velocidad para medir el progreso.

En el ámbito de trabajo de este trabajo de fin de máster se considerarán los siguientes métodos ágiles:

- **Kanban** [8]
- **Lean Development** [9]
- **Extreme Programming (XP)** [10]
- **Scrum** [11]

Implantación

Como ya se ha comentado anteriormente la transición al agilismo es una tarea compleja y difícil de realizar, pero el esfuerzo requerido para realizar la implantación se verá recompensado ampliamente (mayor productividad y menor coste, mejor compromiso y satisfacción del empleado al realizar su trabajo, mejora en la satisfacción del cliente, etc.).

El trabajo que se debe realizar para una correcta implantación de metodologías ágiles se encapsula en las siguientes 5 actividades [12]:

- **Conciencia** de que las metodologías tradicionales no ofrecen unos resultados aceptables.
- **Deseo** de adoptar metodologías ágiles para solucionar así los problemas actuales.
- **Habilidad** de tener éxito con métodos ágiles.
- **Promoción** de las prácticas ágiles a través del compartimiento de experiencias para que otros puedan ver nuestro éxito.
- **Transferencia** de las implicaciones del uso de prácticas ágiles por toda la empresa.

¿Pero cómo llevar a cabo la implantación de prácticas ágiles? ¿Se deberían aplicar todas al mismo tiempo o de forma incremental? La mayoría de expertos aconsejan una adopción de forma incremental (ver Figura 3), ya que esta aproximación conlleva menor riesgo y una alta probabilidad de éxito, en la que las empresas poco a poco irán evolucionando hacia el agilismo.

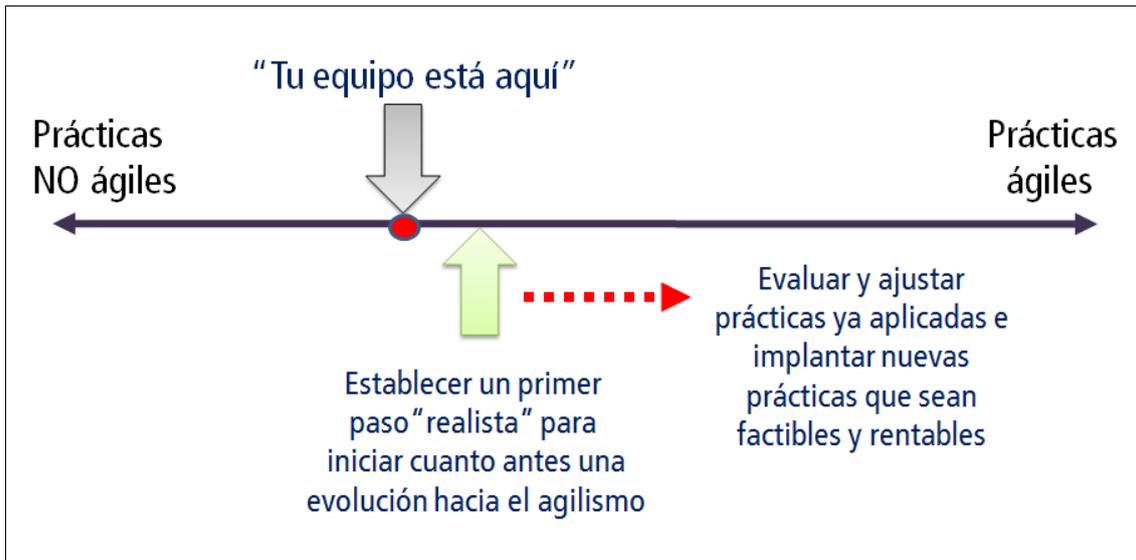


Figura 3. Evolución hacia el agilismo [1]

Esta aproximación será menos costosa para la empresa, en donde un primer paso sería seleccionar un subconjunto de prácticas ágiles que deberían de ir aplicando en sus equipos de trabajo. Una ventaja clara que ofrece esta aproximación es que sería menos estresante para los trabajadores de la organización, ya acostumbrados a trabajar con una metodología tradicional. Además, se podrán realizar pequeños cambios hacia el agilismo sin la necesidad de reorganizar la estructura de la empresa, aunque ésta deberá de cambiar eventualmente.

La implantación de prácticas ágiles deberá ser ágil también, en donde las organizaciones dispondrán de una (o más, según el número de equipos y tamaño de la empresa) hoja de ruta para la implantación de prácticas ágiles. Al igual que en Scrum se utiliza un *product backlog*, esta hoja de ruta vendría a ser un *improvement backlog*, en donde quedaría recogido todas las prácticas que la organización desee adoptar. [13]

Esta forma de proceder puede ser muy interesante para las empresas que deseen empezar a implantar aquellas prácticas que no entren en conflicto con estándares ya establecidos, como pueden ser la ISO 9001 (en inglés, *The International Organization for Standardization*) o el CMMI (en inglés, *Capability Maturity Model Integration*).

Se constata entonces que es necesario realizar un diagnóstico y contar con una herramienta que nos asista en el proceso de implantación de prácticas ágiles. En ella una empresa podrá valorar en qué estado se encuentra y poder así construir su propia hoja de ruta de implantación de prácticas ágiles.

4. Estado del Arte

Actualmente existen herramientas en el mercado que te permiten diagnosticar cuán ágil es una organización. Sin embargo, no existe una herramienta que permita asistir la implantación de prácticas ágiles en los equipos de trabajo de forma incremental. De ahí que surgiera la herramienta Agile Roadmap [14], y posteriormente, como objetivo de esta tesis Agile Roadmap+.

Se comentan a continuación algunas de las herramientas existentes con un propósito cercano al nuestro.

Comparative Agile Assessment (CA)

<http://comparativeagility.com/>



El contenido de este apartado ha sido resumido desde la referencia [15].

Herramienta creada por Mike Cohn, Kenny Rubin y Laurie Williams de uso gratuito. Asiste a una organización a diagnosticar cuán ágil es una organización en comparación a tus competidores. Centrada en la idea de que un negocio no ha de ser perfecto, sino que mejor que sus rivales en el mercado (e.g. Google es el buscador dominante actualmente no porque sus resultados sean perfectos, sino porque suelen ser mejor que su competencia).

Mediante una serie de preguntas que se realizan a la empresa, se comparan sus resultados con la base de datos para comprobar a qué nivel se encuentran con respecto a la competencia. También se pueden comparar los resultados con un subconjunto de la base de datos (e.g. empresas que trabajen en desarrollo web).

CA diagnostica el agilismo de una empresa en 7 dimensiones:

- Trabajo en equipo
- Requerimientos
- Planificación
- Prácticas ágiles
- Calidad
- Cultura
- Creación de conocimiento

En la figura 4 se muestran resultados parciales en tres dimensiones de un equipo de trabajo. En la figura se visualizan cuán ágil es el equipo en comparación a otros, que han sido extraídos de la base de datos (en este caso, equipos de trabajo que realizan desarrollo web). Un valor de 0 representa el valor medio de todos los equipos. Las líneas verticales etiquetadas de -2 a 2 representan la desviación típica con respecto a la media. Se puede observar que en este caso el equipo lo está haciendo muy bien en planificación, un poco mejor que la media en requerimientos y bastante peor que los demás en calidad.

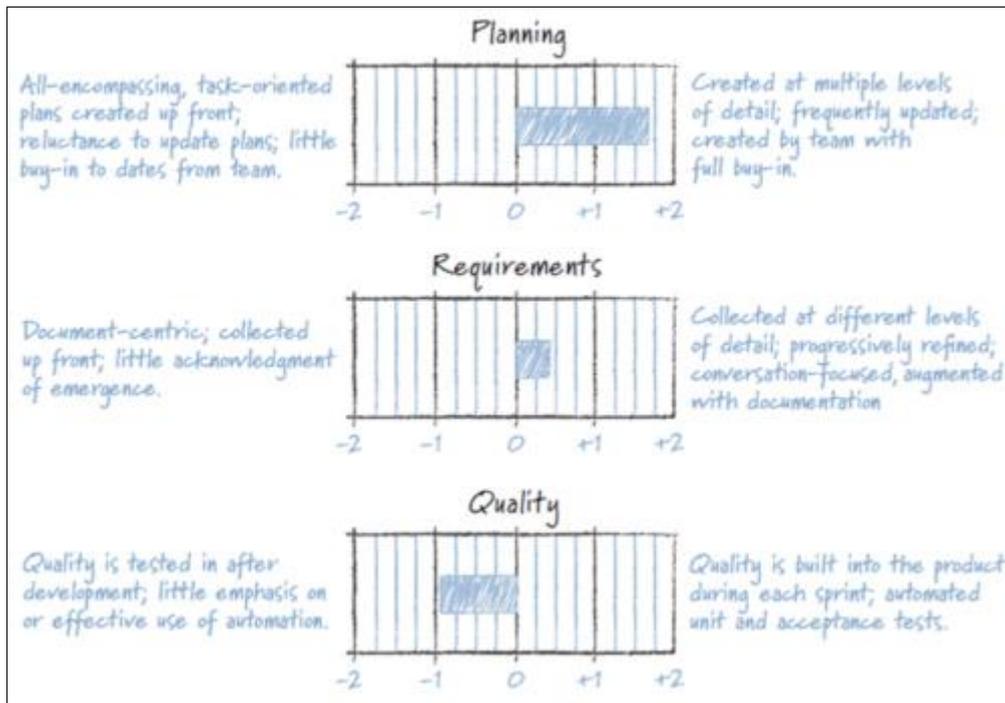


Figura 4. Ejemplo Comparative Agility

Cada una de las tres dimensiones mostradas está compuesta de entre tres a seis características. Un conjunto de preguntas es realizado para calcular la puntuación del equipo en cada característica. Por ejemplo, en el caso de la dimensión de planificación, se realizan las siguientes preguntas (ver Figura 5) para evaluar la característica de “cuándo ocurre la planificación”.

	True	More True than False	Neither True Nor False	More False than True	False	Not Applicable
Up-front planning is helpful without being excessive.						
Team members leave planning meetings knowing what needs to be done and have confidence they can meet their commitments.						
Teams communicate the need to change release date or scope as soon as the need is discovered.						
Effort spent on planning is spread approximately evenly throughout the project.						

Figura 5. CA - Cuándo ocurre la planificación

Esta herramienta permite diagnosticar cómo de ágil es tu empresa o tu equipo de trabajo con respecto a la competencia. Pero una vez realizada la evaluación, carece de una guía clara de qué prácticas se deberían de implantar para mejorar estos resultados.

Agile Maturity Model (AMM)

El siguiente apartado es un resumen de [16].

Scott Ambler, mediante un modelo compuesto por cinco estados, describe cómo de ágil se encuentran las organizaciones en cada una de ellos.

El conjunto de estados es el siguiente:

- *Rethorical stage*: En este estado los equipos han comenzado con el agilismo, pero todavía tienen un largo camino por recorrer.
- *Certified stage*: En este estado las organizaciones se encuentran certificadas por entidades externas, a pesar de que el autor comenta la poca utilidad de ello.
- *Plausible stage*: En este estado se centra el foco de atención en las posibles estrategias ágiles que pueden ser viables en el contexto de la organización.
- *Respectable stage*: En este estado los equipos ya han adoptado un método ágil de forma completa.
- *Measured stage*: En este estado la organización recoge métricas en tiempo real para poder realizar predicciones basadas en datos empíricos.

Su objetivo principal es mejorar la efectividad en el desarrollo ágil en los productos software.

La utilización de este modelo es una buena forma de diagnosticar en qué estado se encuentra una empresa o equipo de trabajo actualmente. Por otra parte, al igual que la herramienta anterior, no se ofrece una funcionalidad que asista a los equipos de trabajo a decidir qué prácticas ágiles se habrían de adoptar, teniendo como objetivo mejorar su nivel de agilidad.

Enterprise Agility Maturity Matrix

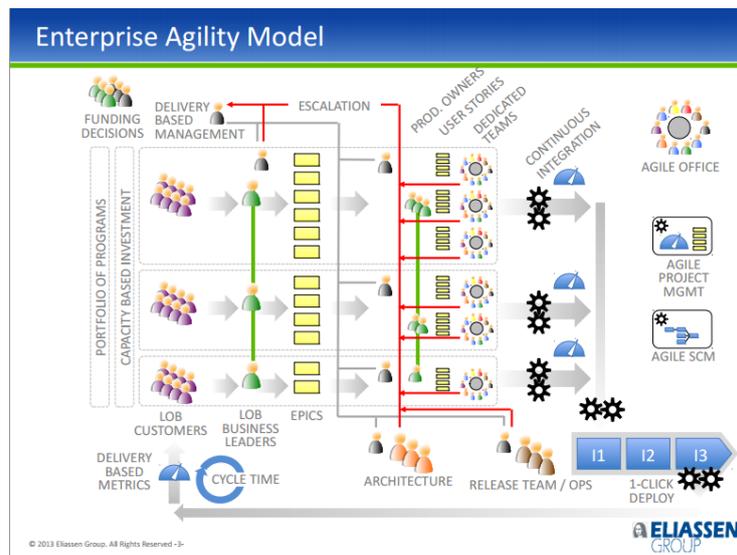


Figura 6. Agile Maturity Model

El siguiente fragmento es un resumen de [17].

Esta herramienta está compuesta por una hoja de cálculo con una sección que describe la organización como un todo y otra para describir los equipos de forma individual. Existen una serie de indicadores ágiles para cada sección y cada uno oscila desde '0' (obstaculizado) hasta '4' (ideal). Por cada celda en la hoja de cálculo se encuentra una explicación de qué significa estar a ese nivel con un indicador determinado. El objetivo es conseguir el mayor número de indicadores con un valor de '2' (sostenible). Todo esto ayuda a la organización entender cuándo han sobrepasado el mayor obstáculo en la adopción de prácticas ágiles y que así no se interrumpa la inversión en la adopción demasiado pronto.

Conclusión

Las herramientas revisadas no apoyan en la evaluación y selección de prácticas ágiles para un contexto determinado, sino más bien permiten saber en qué nivel de agilismo se encuentra una organización.

5. Agile Roadmap+

Punto de partida. Agile Roadmap.

“Un Agile Roadmap es la hoja de ruta para la implantación de prácticas ágiles en un equipo de trabajo. Es equivalente a un *Improvement Backlog* para un equipo de trabajo, es decir, una lista ordenada de mejoras de proceso que se quiere ir implantando a lo largo del tiempo en el equipo de trabajo. Un Agile Roadmap implícitamente reconoce el hecho que es difícil y muchas veces no recomendable implantar muchas prácticas a la vez y en toda su profundidad. Por esto es útil tener una lista ordenada de prácticas candidatas que podrían irse implantando paulatinamente. También implícitamente un Agile Roadmap centra el foco en la aplicación de prácticas, NO en conseguir aplicar una metodología ágil en particular.” [18]

En el trabajo de Fin de Máster de Fausto Nelson (dirigido por Patricio Letelier) [14], se realizó una primera versión de un sitio web en el que se ayudaba al interesado en generar su propio Agile Roadmap, según las necesidades de su organización. El sitio web generaba una lista de prácticas ágiles a implantar en el orden más conveniente. Sin embargo, se detectaron algunas carencias en dicho sitio, siendo las siguientes las más notorias:

- Falta de flexibilidad al realizar el roadmap. La construcción del roadmap estaba sujeto a una fórmula concreta, y en ella era muy difícil considerar la gran diversidad existente entre las numerosas organizaciones.
- Usabilidad pobre. El sitio era poco usable y no se adaptaba a dispositivos móviles.
- El roadmap se había de realizar en una única sesión ya que no se guardaba el estado de éste.

Agile Roadmap+

Dadas las carencias en el sitio web anterior se propuso la creación de un sitio alternativo al original. A diferencia del Agile Roadmap original, en este nuevo sitio se dispone ya de un ruta por defecto de prácticas ágiles a implantar en una organización, cuyo orden se podrá modificar según las necesidades específicas del interesado (una vez ya se esté registrado). Como mejora al sitio anterior destacar que se guardará el estado de los cambios realizados, con lo que no será necesario construir un Agile Roadmap en una sola iteración. Para mejorar la usabilidad el sitio se rediseñará desde cero y se aplicará tecnología puntera para desarrollo web.

Modelo

El modelo utilizado (ver Figura 7) para la construcción de la herramienta es el mismo al utilizado en la herramienta anterior, en cambio, ahora la explotación será radicalmente diferente, flexibilizando la forma en la cual se evalúan las prácticas.

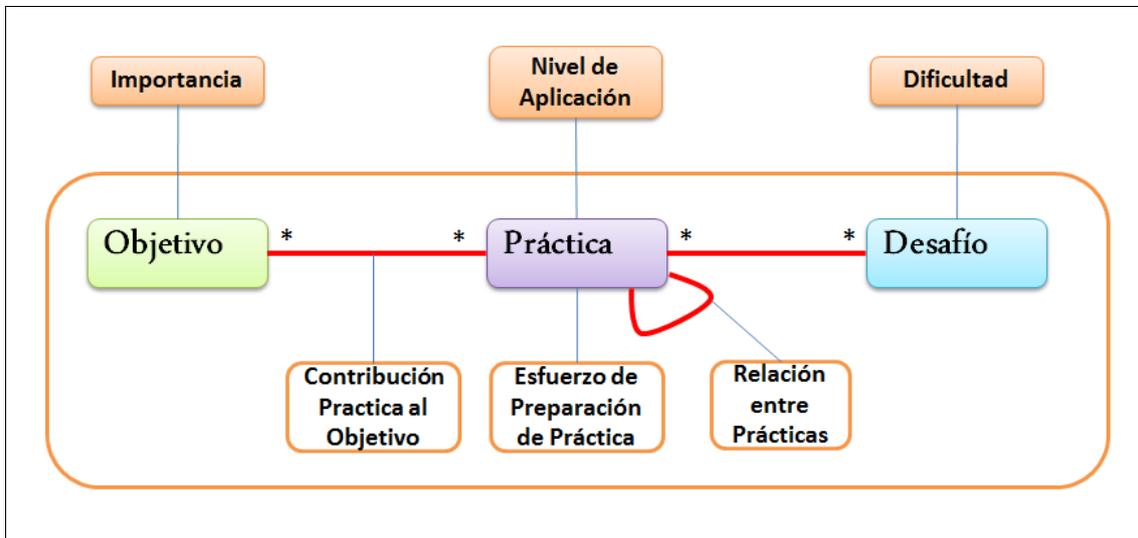


Figura 7. Modelo AgileRoadmap

El modelo se basa principalmente en la lista de 42 prácticas ágiles a implantar en una organización. En cada una de las prácticas se ha de evaluar su nivel de aplicación, el esfuerzo de implantación y si interesa su implantación. También se detalla qué prácticas contribuyen a unos determinados objetivos, los desafíos planteados por cada una de ellas y las relaciones existentes entre cada práctica.

Prácticas Ágiles

En el modelo se incluyen 42 prácticas ágiles (ver Figura 8) que podrían implantarse en equipos de trabajo. Esta lista de prácticas ágiles ha sido elaborada por Patricio Letelier [19]. La mayoría de estas prácticas han sido extraídas de los métodos ágiles más populares (Kanban, Lean Development, Scrum y Extreme Programming), y otras, son sacadas de su propia experiencia.

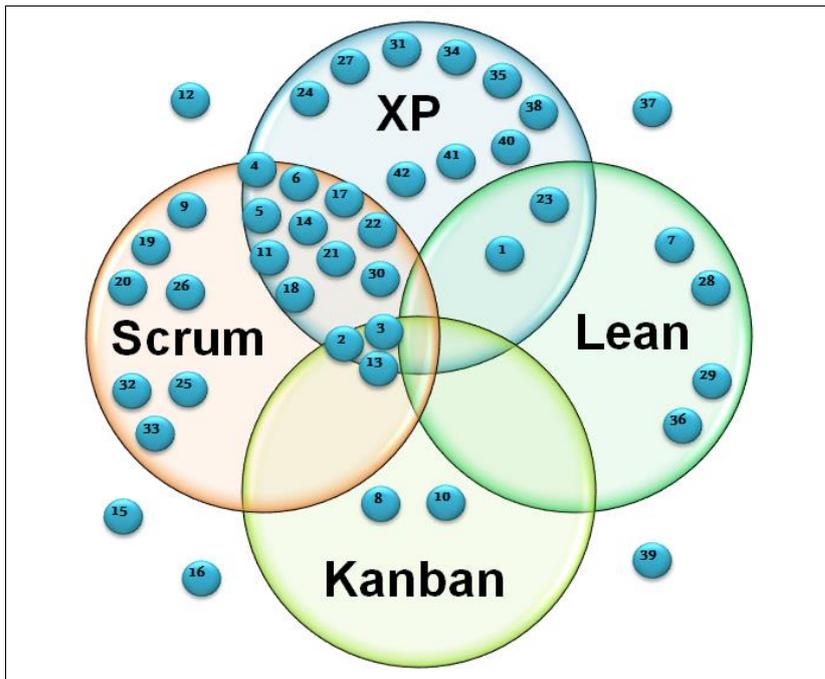


Figura 8. Agrupación por método ágil

La Figura 8 ilustra que existen prácticas comunes y exclusivas de métodos, con lo que no tiene mucho sentido obsesionarse con implantar un determinado método.

El propósito es aplicar el mayor número de prácticas ágiles, y no el de aplicar un método en concreto.

Objetivos Específicos de Mejora

Junto con las 42 prácticas, se han recolectado 16 objetivos específicos de mejora, las cuales se agrupan en tres categorías:

- Satisfacción del cliente.
- Motivación y compromiso del equipo.
- Productividad.

La Figura 9 muestra los 16 objetivos y las categorías asociadas.

La selección de prácticas ágiles debería en primera instancia estar condicionada por los objetivos de mejora que se quieren conseguir. Cada práctica ágil puede contribuir en cierta medida a conseguir un objetivo.

Objetivos específicos de mejora	Satisfacción del Cliente		
	Motivación y compromiso del equipo	Productividad	
Alineación del trabajo del equipo con los objetivos del negocio	X		
Evitar o reducir los retrasos en las entregas	X		
Reducir defectos en el trabajo entregado al cliente	X		
Reducir el tiempo de entrega al cliente, acelerar el "time to market"	X		
Involucrar en mayor medida al cliente en la planificación, definición y validación del trabajo	X		
Tomar decisiones en el momento oportuno	X		
Mejorar la comunicación dentro del equipo y con el cliente	X	X	
Hacer más visible el estado del trabajo del equipo		X	
Reducir las horas extras o demanda no prevista de recursos humanos adicionales		X	
Gestionar eficazmente el contexto multi-proyecto	X	X	X
Evitar costos asociados a la realización de tareas prescindibles o dudosamente rentables		X	X
Reducir el re-trabajo debido a trabajo defectuoso o incompleto detectado por el equipo		X	X
Gestionar eficazmente los cambios, tanto en los trabajos como en sus prioridades		X	X
Promover la mejora continua del proceso empleado por el equipo		X	X
Mejorar la gestión de recursos humanos en el equipo		X	X
Mejorar la sistematización del trabajo		X	X

Figura 9. Objetivos específicos de mejora

Desafíos Implantación Prácticas

Desafíos asociados al implantar cada una de las prácticas. La implantación de cada práctica será satisfactoria si se consiguen superar los desafíos planteados en cada una de ellas.

Características Principales sitio Agile Roadmap+

Entraremos ahora en detalle con las características más importantes de este nuevo sitio web.

Catálogo de Prácticas Ágiles

Sin necesidad de estar registrado se podrá acceder a la lista de 42 prácticas ágiles para implantar en equipos de trabajo. En la Figura 10 se muestra cómo se visualiza la lista de prácticas.

Catálogo de Prácticas Ágiles			
Número	Nombre de la Práctica Ágil	Método Ágil	Esfuerzo
1	Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente.	Lean, XP	Bajo
2	Abordar y entregar trabajo terminado de forma incremental.	Kanban, XP, Scrum	Alto
3	Realizar entregas frecuentes de unidades de trabajo terminadas	Kanban, XP, Scrum	Muy alto
4	Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses).	XP, Scrum	Bajo
5	Acotar el trabajo previsto para un período en base a su estimación y la correspondiente coherencia con la capacidad del equipo	XP, Scrum	Muy alto
6	Organizar el trabajo en iteraciones que agrupan unidades de trabajo que son entregadas en una fecha prevista.	XP, Scrum	Medio
7	Evitar invertir esfuerzo en adelantar trabajo que no esté comprometido y/o no esté cercano a su entrega.	Lean	Muy poco

Figura 10. Catálogo Prácticas

Se podrá consultar toda la información relativa a cada práctica (nombre, descripción, metodología ágil de la que proviene, esfuerzo implantación). En la figura 11 se puede ver cómo se muestra una práctica.

<p><u>Nombre</u></p> <p>Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente.</p> <p><u>Descripción</u></p> <p>Ejemplo de situación con la anti-práctica es el siguiente: el cliente y el equipo se esmeran en conseguir una primera versión del producto demasiado ambiciosa en cuanto a las características que incluye y/o en cuanto al nivel de complejidad o sofisticación en el que se ofrecerán algunas de dichas características.</p> <p>Hasta que el producto no se comience a utilizar no se tendrá una apreciación precisa del nivel de uso de las características del producto y de la utilidad que ellas ofrecen. Para evitar desperdicio de esfuerzo en desarrollo de características poco utilizadas o subutilizadas respecto de toda su funcionalidad implementada es preferible limitarse (al menos durante la generación de un producto) a trabajar con el diseño más sencillo que resuelve la necesidad del cliente y con el mínimo conjunto de características que pueden constituir un producto útil para el cliente. Esta práctica es mencionada como "Diseño simple" en Extreme Programming, y en Lean Development esta práctica está alineada con el principio de eliminación de desperdicios, en este caso referido a trabajo que es prescindible y/o que no es valorado por el cliente. Un término muy en sintonía con esta práctica es el de Minimum Viable Product (MVP), u otro también similar llamado Minimum Marketable Features (MMF), los cuales se refiere al conjunto más pequeño posible de características del producto o servicio que por sí mismas podrían aportar valor, constituyendo por ejemplo, una primera entrega de un producto o una primera versión de una nueva funcionalidad de cierta envergadura. En una estrategia similar encontramos el Principio KISS ("Keep it simple, stupid") o el acrónimo YAGNI ("You aren't gonna need it"), el primero insiste en no sofisticar las soluciones y el otro en contener la ambición de añadir en el momento actual elementos no imprescindibles, y que incluso en el futuro podrían no ser necesarios. Otro aspecto interesante y relacionado con esta práctica se menciona entre los 7 Mudas (fuentes de desperdicio) de Lean Manufacturing, enunciado como: "Sobre-proceso: Trabajo o servicio adicional no percibido por el cliente", es decir, un esfuerzo invertido que no llega a ser valorado por el cliente.</p> <p><u>Método Ágil</u></p> <p>Lean, XP</p>

Figura 11. Detalle Práctica

Agile Roadmap+

En esta sección se presentan las explicaciones de Agile Roadmap y Agile Roadmap+, además de los pasos a realizar para construir un Agile Roadmap+. También se tendrá acceso al catálogo de prácticas anteriormente comentado y a los integrantes del equipo encargados de construir el sitio.

Mi Agile Roadmap+

Función principal de la aplicación, en donde el usuario una vez ya registrado podrá construir su propio Agile Roadmap.

Ordenamiento Prácticas

Mediante un simple *drag&drop* el usuario podrá alterar qué posición tendrá una determinada práctica en la hoja de implantación. En la figura 12 y en la figura 13 se ilustra el inicio y el fin del *drag&drop*.

Roadmap de Prácticas Ágiles

Ocultar prácticas sin margen de mejora
 Ocultar practicas no aplicables

	Orden	Nombre de la Práctica Ágil	Método Ágil	Esfuerzo Implantación	Margen de Mejora	Notas	No aplicable
	1	PRA1 🐜 Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente.	Lean, XP	Bajo	Medio	polgkug	<input type="checkbox"/>
	2	PRA3 🐜 Realizar entregas frecuentes de unidades de trabajo terminadas	Kanban, XP, Scrum	Muy alto	Ninguno		<input type="checkbox"/>
	3	PRA4 🐜 Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses).	XP, Scrum	Bajo	No definido		<input type="checkbox"/>

Figura 12. Inicio del dragdrop

Roadmap de Prácticas Ágiles

Ocultar prácticas sin margen de mejora
 Ocultar practicas no aplicables

	Orden	Nombre de la Práctica Ágil	Método Ágil	Esfuerzo Implantación	Margen de Mejora	Notas	No aplicable
	1	PRA3 🐜 Realizar entregas frecuentes de unidades de trabajo terminadas	Kanban, XP, Scrum	Muy alto	Ninguno		<input type="checkbox"/>
	2	PRA1 🐜 Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente.	Lean, XP	Bajo	Medio	polgkug	<input type="checkbox"/>

Figura 13. Fin del drag&drop

Evaluación de Prácticas

El usuario por cada práctica podrá especificar el margen de mejora (margen que tiene actualmente la práctica respecto del máximo nivel de aplicación posible), si es aplicable o no (se marcará el *checkbox* cuando no sea aplicable) y podrá añadir comentarios si así lo desea. En la figura 14 se ilustra una práctica en concreto.

	1	PRA3 🐜 Realizar entregas frecuentes de unidades de trabajo terminadas	Kanban, XP, Scrum	Muy alto	Alto	Implantar lo antes posible.	<input type="checkbox"/>
--	---	---	-------------------	----------	------	-----------------------------	--------------------------

Figura 14. Ejemplo de evaluación práctica

Mostrar/Ocultar Prácticas

Al usuario se le dará la opción de mostrar/ocultar aquellas prácticas que, o bien no sean aplicables (*checkbox* marcado), o que como margen de mejora contengan el valor de “Muy Alto”, es decir, que estén totalmente aplicadas.

Ver Prácticas Relacionadas

Por cada práctica el usuario podrá solicitar que se resalten todas las prácticas que estén relacionadas con dicha práctica, haciendo click sobre este icono . Como se ilustra en la figura 15 la práctica seleccionada estará marcada mediante un color azul claro, mientras que las relacionadas estarán resaltadas con un color amarillo.

	Orden	Nombre de la Práctica Ágil	Método Ágil	Esfuerzo Implantación	Margen de Mejora	Notas	No aplicable
+	1	PRA3 🛠 Realizar entregas frecuentes de unidades de trabajo terminadas 	Kanban, XP, Scrum	Muy alto	Alto	implantar lo antes posible.	<input type="checkbox"/>
+	2	PRA1 🛠 Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente. 	Lean, XP	Bajo	Ninguno	polgkug	<input type="checkbox"/>
+	3	PRA4 🛠 Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses). 	XP, Scrum	Bajo	No definido		<input type="checkbox"/>
+	4	PRA2 🛠 Abordar y entregar trabajo terminado de forma incremental. 	Kanban, XP, Scrum	Alto	No definido		<input checked="" type="checkbox"/>
+	5	PRA5 🛠 Acotar el trabajo previsto para un período en base a su estimación y la correspondiente coherencia con la capacidad del equipo 	XP, Scrum	Muy alto	No definido		<input type="checkbox"/>

Figura 15. Ver Prácticas Relacionadas.

Mostrar los Desafíos de una Práctica

Se podrán mostrar los desafíos que supone llevar a cabo la implementación de cada práctica. Haciendo click sobre el icono se accederá a los desafíos de la práctica correspondiente (ver Figura 16).

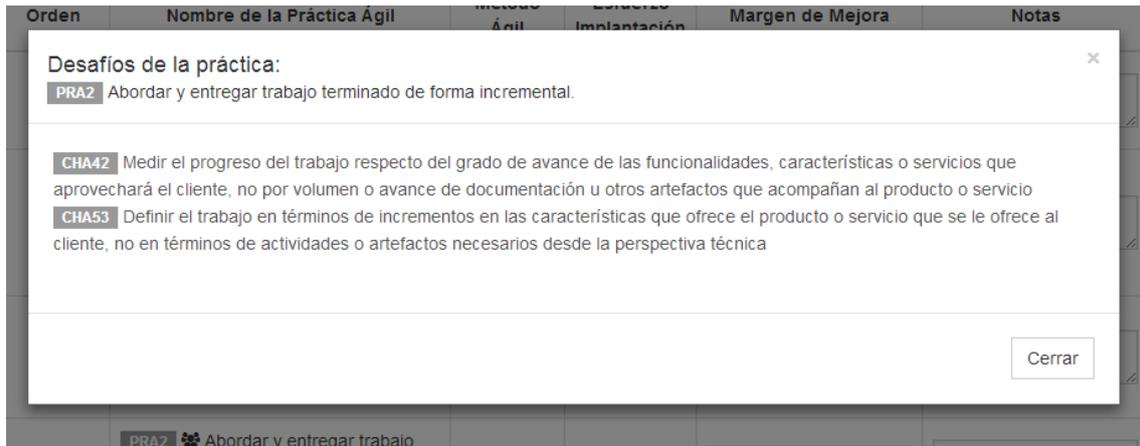


Figura 16. Desafío Práctica

Objetivos de Mejora

Los objetivos de mejora se podrán ordenar mediante *drag&drop* de forma idéntica a como se realiza con el ordenamiento de las prácticas. La Figura 17 muestra cómo se visualiza la lista de objetivos.

Objetivos de mejora					
	Orden	Objetivo específico de mejora	Satisfacción del cliente	Motivación y compromiso del equipo	Productividad
+	1	OBJ11 Alineación del trabajo del equipo con los objetivos del negocio	✓		
+	2	OBJ1 Evitar o reducir los retrasos en las entregas	✓		
+	3	OBJ5 Reducir defectos en el trabajo entregado al cliente	✓		
+	4	OBJ21 Reducir el tiempo de entrega al cliente, acelerar el "time to market"	✓		
+	5	OBJ12 Involucrar en mayor medida al cliente en la planificación, definición y validación del trabajo	✓		
+	6	OBJ8 Tomar decisiones en el momento oportuno	✓		
+	7	OBJ10 Mejorar la comunicación dentro del equipo y con el cliente	✓	✓	
+	8	OBJ7 Hacer más visible el trabajo del equipo		✓	
+	9	OBJ4 Reducir las horas extras o demanda no prevista de recursos humanos adicionales		✓	
+	10	OBJ22 Gestionar eficazmente el contexto multi-proyecto	✓	✓	✓

Figura 17. Objetivos de mejora

La funcionalidad más interesante es la de poder visualizar por cada objetivo qué prácticas ayudan a conseguir dicho objetivo, indicando así mismo su contribución (Muy Baja, Baja, Media, Alta, Muy Alta) tal como se muestra en la figura 18.

	Orden	Nombre de la Práctica Ágil	Método Ágil	Esfuerzo Implantación	Margen de Mejora	Notas	No aplicable	Contribución del objetivo
+	1	PRA3 🐛 Realizar entregas frecuentes de unidades de trabajo terminadas 🔍📄👁	Kanban, XP, Scrum	Muy alto	Alto	Implantar lo antes posible.	<input type="checkbox"/>	
+	2	PRA1 🐛 Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente. 🔍📄👁	Lean, XP	Bajo	Ninguno	polgkug	<input type="checkbox"/>	
+	3	PRA4 🐛 Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses). 🔍📄👁	XP, Scrum	Bajo	No definido		<input type="checkbox"/>	Baja
+	4	PRA2 🐛 Abordar y entregar trabajo terminado de forma incremental. 🔍📄👁	Kanban, XP, Scrum	Alto	No definido		<input checked="" type="checkbox"/>	Baja

Figura 18. Contribuciones de las prácticas ágiles a un objetivo

Proceso de Desarrollo

Se ha utilizado TUNE-UP Processs (<http://www.tuneupprocess.com/>) durante el proceso de desarrollo. El autor desempeñaba el rol de desarrollador, mientras que mi tutor, Patricio Letelier, se encargaba de las pruebas de aceptación. En la definición de las unidades de trabajo participamos ambos. El desarrollo de la herramienta Agile Roadmap+ comenzó a finales de Octubre de 2013 y ha finalizado en Junio de 2014, aunque ya desde de Abril de 2014 la aplicación está disponible. En la figura 19 se puede observar que se ha trabajado en más de 40 unidades de trabajo (UTs).

Tipo	Orden	UT	Nombre UT	Importancia	Urgencia	Riesgo	Esfuerzo a priori
★	5	2964	Quitar el icono indicador de movimiento de fila en la cabecera de listas de objetivos y de	Baja	Baja	Muy Bajo	0
★	100	2940	Botón deshacer último cambio de posición	Baja	Muy Baja	Bajo	1
★		3103	Mejoras en sitio	Media	Alta	Bajo	1
★		2972	Columnas en diferentes tamaños de pantalla	Media	Media	Muy Bajo	0.5
★	1	2963	Usar icono de ojo tapado para "Ver lista de objetivos" "No ver lista de objetivos", poniendo	Media	Media	Bajo	0.5
★	1	2973	Cambiar y uniformar la presentación de códigos	Media	Media	Muy Bajo	0.5
★	10	3233	Quitar texto ...	Media	Media	Muy Bajo	0
★	20	2939	Mecanismo para ocultar prácticas no aplicables o sin margen de mejora	Media	Media	Medio	3
★	20	3225	En la lista de prácticas ocultar las columnas Número y Esfuerzo, y poner como prefijo del	Media	Media	Muy Bajo	1
★	30	2858	Registro de usuario	Media	Media	Medio	1
★	70	3223	Quitar los apartados método ágil y esfuerzo en las descripciones de prácticas	Media	Media	Muy Bajo	0.5
★	100	3175	Almacenar para cada usuario sus marcar de los checks que ocultan filas	Media	Alta	Bajo	1
★		3117	Retosques de IU	Alta	Alta	Bajo	2
★		3118	Campo en registro e IP	Alta	Alta	Bajo	1
★		3119	Limpieza y usuarios de prueba	Alta	Alta	Bajo	1
★	10	2938	Interfaz para correlacionar objetivos con prácticas	Alta	Media	Medio	5
★	10	2965	Intentar mejorar la visualización de los iconos	Alta	Media	Bajo	1
★	10	2857	Montar Sitio	Alta	Media	Medio	3
★	10	2854	Gestión de varios Roadmaps en una misma cuenta	Alta	Media	Medio	2
★	18	2856	Mapa de Prácticas Ágiles	Alta	Media	Medio	3
★	20	2978	Cambios en páginas del sitio	Alta	Alta	Medio	5
★	20	2855	Gestión del RoadMap de Organización	Alta	Media	Medio	5
★	50	2976	SQL Server 2008 setup	Alta	Media	Medio	5
★	80	3224	Añadir un icono que muestre los objetivos a los que contribuye una práctica y en qué medida	Alta	Media	Bajo	2
★	90	3176	Añadir funcionalidad de ocultar objetivos con check "Ocultar objetivos que NO interesan"	Alta	Media	Bajo	1
★	110	3177	Ofrecer columna Notas en objetivos	Alta	Media	Bajo	1
★	50	3178	Poder marcar un objetivo o una práctica como ya revisada	Muy Alta	Alta	Medio	2
★		3104	Detalles en contenido de sitio	Muy Baja	Media	Medio	

Figura 19. Unidades de Trabajo

Cada unidad de trabajo correspondía a una de las siguientes características presentes en la aplicación:

- Página Principal
- Login
- Registro de Usuario
- Mapa de Prácticas Ágiles
- Roadmap
- Mis Empresas
- Estadísticas

En cuanto al flujo de trabajo, se puede observar en la siguiente gráfica que la mayor parte de unidades de trabajo se completaron entre los meses de Febrero y Marzo.

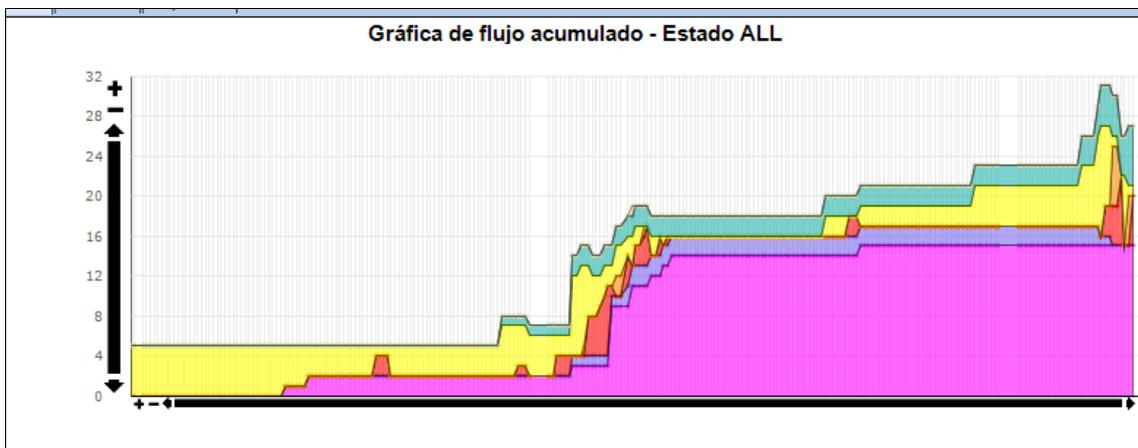


Figura 20. Flujo de Trabajo

Durante todo el periodo de desarrollo, aproximadamente se han invertido unas 20 horas en definición de unidades de trabajo, 300 horas en programación y unas 40 horas en testeo.

6. Ecosistema tecnológico usado en Agile Roadmap+

La aplicación web se ha implementado usando el framework Ruby on Rails.

Al no disponer de un diseñador gráfico, se ha comprado una plantilla web que utiliza Bootstrap, el framework más popular para el *front-end* desarrollado por Twitter. Aquí podrá encontrarse el diseño original:

<https://wrapbootstrap.com/theme/unify-responsive-website-template-WB0412697>

La plantilla tuvo un coste de 10 euros en el momento de su compra.

Para el control de versiones se ha utilizado Git, usando GitHub como repositorio central.

Se ha utilizado Heroku como entorno de desarrollo, mientras que la puesta en producción se ha realizado en un servidor Ubuntu proporcionado por DigitalOcean, en éste se ha instalado el PaaS Dokku para proporcionar un entorno similar a Heroku.

A continuación se explicarán brevemente cada una de las tecnologías utilizadas.



Ruby on Rails

<http://rubyonrails.org/>

Se ha usado la versión 4 para la implementación de la aplicación. A continuación se comentarán los componentes y tecnologías incorporadas en este framework.

Ruby

<https://www.ruby-lang.org>

El contenido de este apartado viene resumido en [20].

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Según su creador (Yukihiro Matsumoto) Ruby está diseñado para la productividad y la diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario. Sostiene que el diseño de sistemas necesita enfatizar las necesidades humanas más que las de la máquina.

RubyGems

<https://rubygems.org/>

Excelente servicio donde podrás encontrar infinidad de plugins o librerías muy útiles para el desarrollo en Rails. A estas librerías se les denomina usualmente como gemas.

Pasemos ahora a comentar algunas de las gemas más importantes usadas en la aplicación.

Haml

<http://haml.info>

Haml (en inglés, *HTML Abstraction Markup Language*) es el lenguaje de marcado utilizado para definir las vistas (archivos con extensión “.html.haml”), en lugar de usar Ruby embebido (ERB). Además de tener un aspecto más limpio y elegante Haml acelera y simplifica la creación de plantillas.

En el pequeño ejemplo de la figura 21 extraído de la web oficial (<http://haml.info/>) se puede observar el cambio de ERB a Haml.

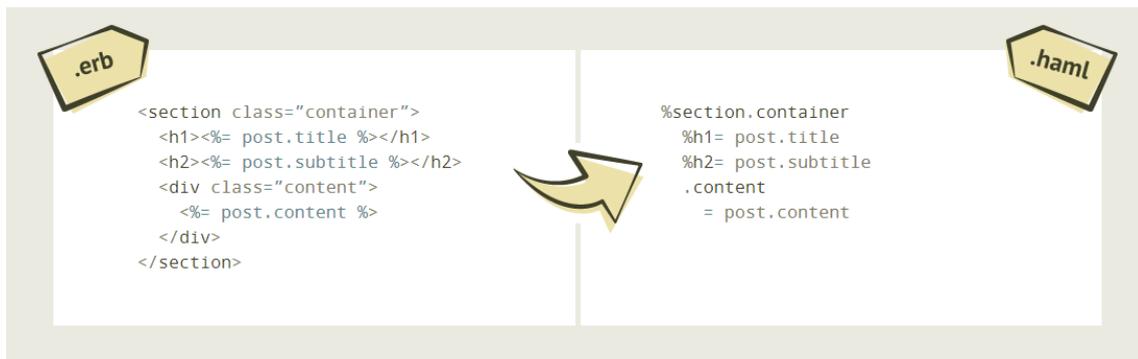


Figura 21. ERB - Haml

Sass

<http://sass-lang.com/>

Gema por defecto en Rails 4. Sass es una extensión de CSS3 que añade anidamiento, variables y mixins entre otras funcionalidades. Posee dos tipos de sintaxis, la que se usará será “SCSS”, ya que en ésta todo código en CSS3 también será válido. El framework se ocupará de traducir el código en Sass a CSS3.



<http://coffeescript.org/>

El contenido de este apartado es un resumen de [21].

Gema también por defecto en Rails 4. CoffeeScript es un lenguaje de programación que se compila a JavaScript. El lenguaje añade azúcar sintáctico inspirado en Ruby, Python y Haskell para mejorar la brevedad y la legibilidad de JavaScript, y añade características más sofisticadas, como la comprensión de listas y la coincidencia de patrones. CoffeeScript compila previsiblemente a JavaScript y los programas se pueden escribir por lo general en 1/3 menos de líneas de código sin ningún efecto en el rendimiento ni en el tiempo de ejecución.



<http://jquery.com/>

El siguiente fragmento es un resumen de [22].

Proporciona jQuery como la librería para JavaScript. jQuery es una librería de JavaScript que facilita la manipulación del DOM (en inglés, “Document Object Model”), animaciones, manejo de eventos y operaciones en Ajax. jQuery ha cambiado la forma en la que millones de personas escriben JavaScript.



<http://jqueryui.com/>

Gema que proporciona todo lo necesario para el uso de la librería jQuery UI. Extiende jQuery con funcionalidades para mejorar la interacción usuario-máquina. Principalmente se ha hecho uso de la funcionalidad *drag&drop*.

Wiselinks

<https://github.com/igor-alexandrov/wiselinks>

Consigue una mayor velocidad en una aplicación gracias al uso de Ajax. Rails usa por defecto *Turbolinks* (<https://github.com/rails/turbolinks>), aunque ofreciendo menos funcionalidades. Ambas son buenas opciones.

Nprogress-rails

<https://github.com/caarlos0/nprogress-rails>

Muestra una pequeña barra de progreso al estilo de YouTube (ver Figura 22) cuando se realiza la navegación mediante Ajax.



Figura 22. Barra progreso Youtube

Touchpunch-rails

<http://touchpunch.furf.com/>

Pequeño *hack* que posibilita la interacción de dispositivos táctiles y la librería jQuery UI.

Font-awesome-sass-rails

<https://github.com/littlebtc/font-awesome-sass-rails>

Librería que facilita la integración de Font Awesome (<http://fontawesome.github.io/Font-Awesome>) en tu aplicación. Font Awesome proporciona un gran número de iconos a usar en tu aplicación de forma gratuita.

Faker

<https://github.com/stympey/faker>

Librería que facilita la generación de información falsa (como nombres, direcciones IP, etc.) de forma automática para facilitar así el testeo de la aplicación.

Chartkick

<http://chartkick.com/>

Permite la creación de gráficos para la visualización de estadísticas de la aplicación. Mediante una única línea de código Ruby se puede crear un gráfico (tecnología JavaScript), por ejemplo para visualizar los registros de usuario por día (ver Figura 23):

```
<%= line_chart User.group_by_day(:created_at).count %>
```

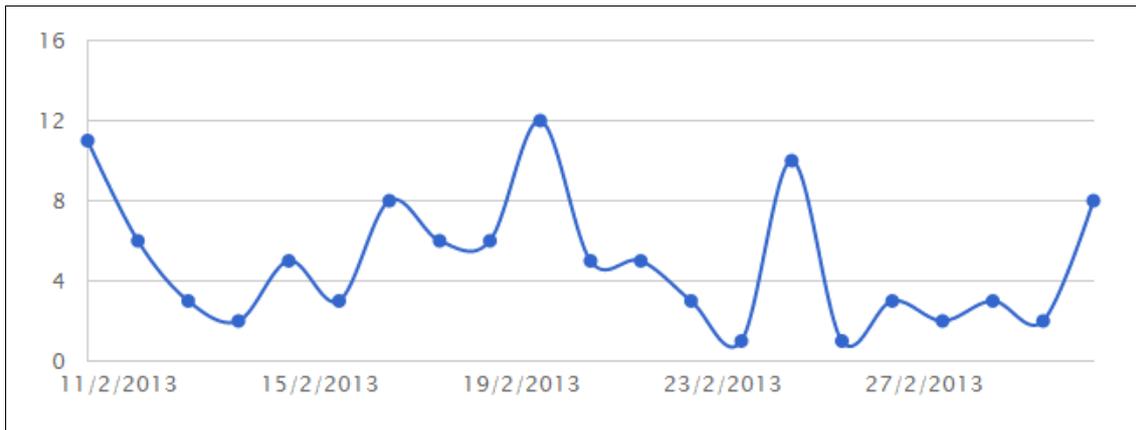


Figura 23. Chartkick

Rails

El contenido de este apartado es un resumen de [23].

Rails es un framework escrito en Ruby para el desarrollo de aplicaciones web. Facilita la programación haciendo suposiciones, lo que permite escribir menos código que en otros lenguajes o frameworks. Además, muchos desarrolladores en Rails aseguran que el uso de este framework es más entretenido.

Rails proporciona un incremento en la productividad por parte del desarrollador gracias a los siguientes principios:

- DRY – Del inglés “Don’t Repeat Yourself” – sugiere que escribir el mismo código una y otra vez es un mal síntoma.
- Convención sobre Configuración (CoC) – significa que Rails decide por ti sobre qué quieres hacer y cómo lo vas a hacer, en vez de tener que especificar cada mínimo detalle en la aplicación.
- REST como el mejor patrón para aplicaciones web – organizando tu aplicación alrededor de tus recursos y verbos HTTP (Get, Post, Put, Delete) es la forma más rápida de proceder.

Instalación de Rails

En <http://railsinstaller.org/en> se ofrece un instalador de Rails con todo lo necesario para el desarrollo en máquinas Windows o Mac (Ruby, Rails, Bundler, Git, Sqlite, etc.).

Alternativas a la instalación

Existen varias alternativas para desarrollar código en la nube que permiten olvidarte de las instalaciones del entorno a utilizar y te permiten escribir código inmediatamente. Entre ellas cabe mencionar:

- Nitrous.io (<https://www.nitrous.io/>): Ofrece entornos para Ruby on Rails, Node.js, Django, Go y PHP. En una sola ventana al alcance de tu navegador

tendrás disponible un editor, consola y chat para colaborar con demás miembros de tu equipo.

- C9.io (<https://c9.io/>): Pensado para proyectos en Node.js.
- Kobra.io (<https://kobra.io/>): Similar a las anteriores, pero además ofrece la posibilidad de video-chat entre los usuarios que desarrollen la aplicación.

Estructura de carpetas

El contenido de este apartado está extraído de [24].

La instalación de Rails crea la siguiente estructura de carpetas.

```
|-- app
|  |-- assets
|     |-- images
|     |-- javascripts
|     `-- stylesheets
|  |-- controllers
|  |-- helpers
|  |-- mailers
|  |-- models
|  `-- views
|     `-- layouts
|-- config
|  |-- environments
|  |-- initializers
|  `-- locales
|-- db
|-- doc
|-- lib
|  `-- tasks
|-- log
|-- public
|-- script
|-- test
|  |-- fixtures
|  |-- functional
|  |-- integration
|  |-- performance
|  `-- unit
|-- tmp
|  |-- cache
|  |-- pids
|  |-- sessions
|  `-- sockets
`-- vendor
    |-- assets
    `-- stylesheets
    `-- plugins
```

A continuación se describe brevemente el contenido de las carpetas.

app

Contiene todo el código específico de la aplicación.

app/assets

Contiene subdirectorios para imágenes, hojas de estilo y archivos JavaScript.

app/controllers

Contiene los controladores que han de ser llamados como “weblogs_controller.rb” para el mapeo automático de URLs. Todos los controladores han de descender de *ApplicationController* que a su vez descende de *ActionController::Base*.

app/models

Contiene los modelos que deben ser llamados como “post.rb”. Descienden de *ActiveRecord::Base* por defecto.

app/views

Contiene las plantillas para las vistas que deben ser llamadas como “weblogs/index.html.erb” para la acción de “WeblogsController#index”.

app/views/layouts

Contiene las plantillas para *layouts* que son usadas conjuntamente con las vistas. En tus vistas, define un *layout* usando `layout :default` y crea un archivo llamado “default.html.erb”. Dentro de dicho archivo, llama a `<% yield %>` para reproducir una vista en este layout.

app/helpers

Contiene *helpers* que proporcionan métodos de ayuda para las vistas.

config

Contiene varios archivos de configuración.

db

Contiene el esquema en “schema.rb”. db/migrate contiene todas las migraciones para tu esquema.

doc

Directorio donde se almacenará toda la documentación de la aplicación.

lib

Librerías específicas de la aplicación. Contiene básicamente cualquier código que no pertenece a controladores, modelos, o helpers.

public

Directorio disponible para el servidor web. Este debe ser el directorio raíz de tu servidor web.

script

Scripts de ayuda para automatización y generación.

test

Test unitarios y funcionales.

vendor

Librerías externas en la que la aplicación depende. También incluye un subdirectorio de plugins.

Base de Datos

Este apartado es un resumen de [25].

En Rails gracias al uso del patrón de diseño *ActiveRecord* el código de la aplicación es independiente de la tecnología de base de datos que se utilice, esto ha posibilitado que para el entorno de desarrollo de la aplicación se ha utilizado SQLite, mientras que para el entorno de producción se ha utilizado PostgreSQL.

ActiveRecord es una aproximación para acceder a la información en una base de datos. Una tabla es envuelta en una clase, de forma que una instancia de ésta corresponde a una fila en la tabla. Después de la creación de un objeto, una nueva fila es añadida a la tabla. Cuando cualquier objeto es cargado obtiene su información de la base de datos, y si es actualizado la fila correspondiente a este objeto es también actualizada. La clase implementa sus métodos de acceso o propiedades para cada una de las columnas en la tabla.

Schema

Este archivo es generado automáticamente por el propio framework. Indica el estado actual de la base de datos. Se encuentra en “db/schema.rb”. Si se quiere cargar el contenido del archivo “schema.rb” en la base de datos del entorno actual, se habría de ejecutar el siguiente comando Rake:

```
>>> rake db:schema:load
```

Seeds

En este archivo se especificarán todos los datos por defecto en la base de datos. Mediante el comando `rake db:seed` se cargarán dichos datos. Se encuentra en “db/seeds.rb”.

En él se especificó todo el conocimiento de la aplicación web (prácticas ágiles, objetivos de mejora, relaciones entre prácticas, desafíos para las prácticas, etc.). De esta forma se podrá replicar la aplicación web con dicho contenido en distintos servidores/entornos de desarrollo.

Migraciones

Si se quiere realizar un cambio en la base de datos, es mala práctica modificar el esquema directamente. Para ello se ha de crear una migración.

Las migraciones son una forma conveniente de alterar la base de datos de una manera estructurada y organizada. *ActiveRecord* reconoce qué migraciones han sido ejecutadas, para actualizar y ejecutar las que no lo hayan sido se procederá con el siguiente comando:

```
>>> rake db:migrate
```

Una de las últimas migraciones creadas en la aplicación fue la creación de una nueva columna “miembros_equipo” para la tabla usuarios. Rails ofrece un comando para generar migraciones de forma rápida y sin la necesidad de crear código para el programador. En este caso fue:

```
>>> rails g migration add_miembros_equipo_to_users
miembros_equipo:integer
```

Pudiendo ver el contenido del código creado por dicho comando en la carpeta ‘db/migrate’.

```
class AddMiembrosEquipoToUsers < ActiveRecord::Migration
  def change
    add_column :users, :miembros_equipo, :integer
  end
end
```

Comandos útiles Rails/Rake

Este apartado está extraído de [26].

A continuación se comentarán algunos comandos de utilidad para Rails. Rake es un *Make* para Ruby, que reemplaza la utilidad “make” de Unix. En Rails, Rake es utilizado para tareas administrativas en donde podrás crear las tuyas propias.

rails server / rails s

Lanza un pequeño servidor web llamado WEBrick que viene incluido en Ruby. Se usará cada vez que se quiera acceder a la aplicación a través de un navegador web. Por defecto utiliza el puerto 3000 (<http://localhost:3000>).

rails console / rails c

Permite interactuar con la aplicación a través de la línea de comandos. Por ejemplo para ver los datos del último usuario registrado:

```
>>> y User.last
```

rails dbconsole / rails db

Automáticamente detecta qué base de datos estás utilizando y te permite interactuar con ella (en nuestro caso SQLite3 para el entorno de desarrollo y PostgreSQL para el entorno de producción).

rails runner / rails r

Ejecuta código Ruby en el contexto de Rails de forma no interactiva. Por ejemplo, como se hizo anteriormente para mostrar los datos del último usuario registrado.

```
>>> rails r "y User.last"
```

rake db:seed

Ejecuta el archivo “db/seeds.rb”. En él se encuentra el contenido inicial de la base de datos.

rake db:migrate

Ejecuta las migraciones que se han de ejecutar en el entorno actual.

rake db:schema:load

Carga el esquema en la base de datos del entorno actual.

rake db:fake_users

Tarea implementada en la que se crean usuarios falsos para poder realizar pruebas con ellos.

rake db:fake_users_generate_info_pracs

Tarea implementada en la que se crea información aleatoria para las prácticas ágiles de los usuarios falsos.



<http://getbootstrap.com/>

Esta sección es un resumen de [27].

Bootstrap es una colección de herramientas de software libre para la creación de sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript.

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Una hoja de estilo llamada `bootstrap.less` incluye los componentes de las hojas de estilo. Bootstrap viene con una disposición de cuadrilla estándar de 940 píxeles de ancho. Alternativamente, el desarrollador puede usar un diseño de ancho-variable. Para ambos casos, la herramienta tiene cuatro variaciones para hacer uso de distintas resoluciones y tipos de dispositivos: teléfonos móviles, formato de retrato y paisaje, tabletas y computadoras con baja y alta resolución (pantalla ancha). Esto ajusta el ancho de las columnas automáticamente.

Dado el auge en los últimos años del uso de dispositivos móviles para el acceso a internet, Bootstrap anima al desarrollador a pensar en un diseño orientado a móviles (en inglés, *mobile first*) [28].

En la aplicación se ha utilizado la versión 3 de Bootstrap. Como se muestra a continuación, gracias a usar un Responsive Design, se ha conseguido que la aplicación se muestre correctamente en todo tipo de pantallas.

Móviles

En la figura 22 se ilustra cómo se visualiza la aplicación web en dispositivos móviles.



Figura 24. Agile Roadmap+ en móviles

Tablets

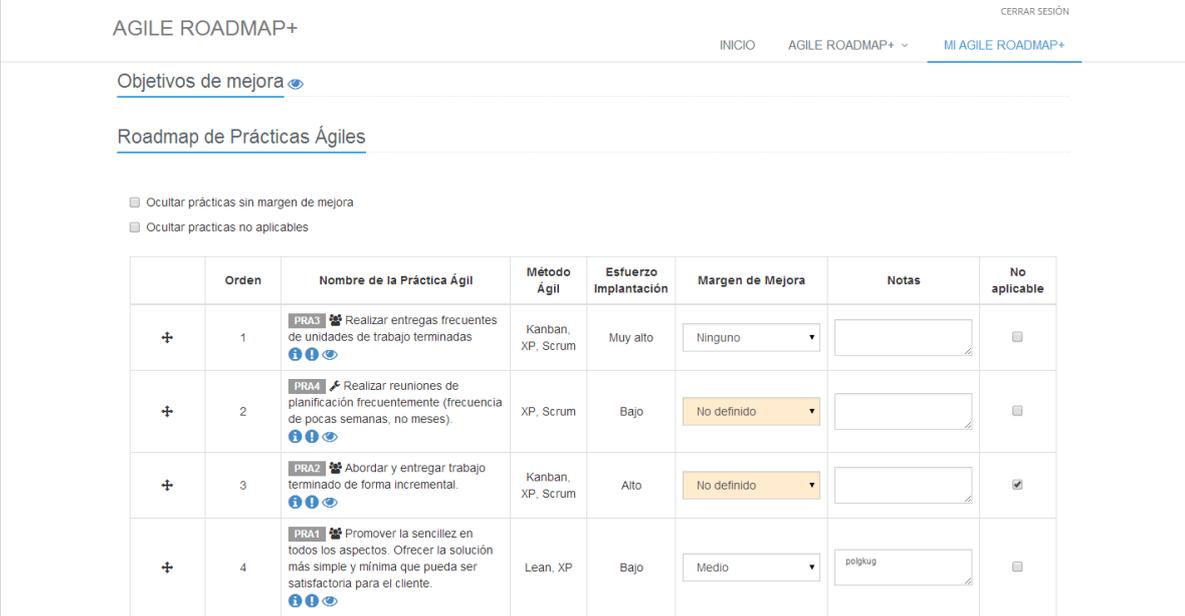
En la figura 23 se ilustra cómo se visualiza la aplicación web en dispositivos móviles de mayor tamaño (tablets).



Figura 25. Agile Roadmap+ en tablets

Desktops

En la figura 24 se ilustra cómo se visualiza la aplicación web en dispositivos con mayor tamaño de pantalla (e.g. ordenadores de sobremesa).



	Orden	Nombre de la Práctica Ágil	Método Ágil	Esfuerzo Implantación	Margen de Mejora	Notas	No aplicable
+	1	PRAS Realizar entregas frecuentes de unidades de trabajo terminadas	Kanban, XP, Scrum	Muy alto	Ninguno		<input type="checkbox"/>
+	2	PRAA Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses).	XP, Scrum	Bajo	No definido		<input type="checkbox"/>
+	3	PRAZ Abordar y entregar trabajo terminado de forma incremental.	Kanban, XP, Scrum	Alto	No definido		<input checked="" type="checkbox"/>
+	4	PRA1 Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente.	Lean, XP	Bajo	Medio	poljlug	<input type="checkbox"/>

Figura 26. Agile Roadmap+ en pc



<http://git-scm.com/>

Este apartado está extraído de [29].

El propósito de Git es gestionar los cambios en un proyecto a lo largo del tiempo. Git almacena esta información en una estructura de datos llamada repositorio. Un repositorio Git contiene entre otras cosas :

- Un conjunto de **commit objects**.
- Un conjunto de referencias a los **commit object**, llamadas **heads**.

Un objeto **commit** está compuesto por:

- Un conjunto de archivos, reflejando el estado del proyecto en un instante en el tiempo.
- Referencias al objeto commit padre.
- Un nombre SHA1, que consiste de una cadena compuesta por 40 caracteres que identifica universalmente el objeto commit.

Cada proyecto siempre tiene un objeto commit sin padres. Este es el primer commit realizado en el proyecto.

Basado en lo anterior, puedes visualizar un repositorio como un grafo acíclico dirigido (DAG) de objetos commit, apuntando cada uno a su padre. De esta forma podrás ver el historial de cambios en el proyecto recorriendo los padres que han llevado a realizar un commit determinado.

Ramas

Esta sección se ha extraído desde [6].

Un concepto muy importante en Git son las ramas o *branches* (ver figura 27). Las ramas nos permiten crear nuevas características en la aplicación sin alterar código que ya funciona.

Una vez la característica esté implementada se podrá entonces hacer merge con la rama principal llamada comúnmente **master**. Si por el contrario decidimos no incluir la característica implementada no habrá ningún problema, simplemente eliminaremos dicha rama.

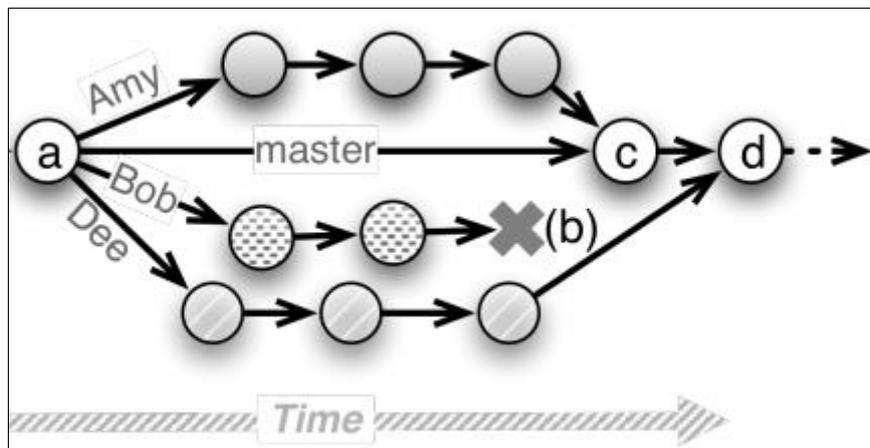


Figura 27. Git ramas

Las ramas deben de tener un periodo de vida corto, de lo contrario se desviarán demasiado de la rama master y será difícil de reconciliar los cambios a realizar. Una buena práctica en Git es no realizar “pequeños” cambios en la rama master por muy simple que parezcan, usar siempre ramas para ello, ya que nunca sabes qué puede salir mal.



GitHub

<https://github.com/>

Servicio que se ha utilizado para alojar el proyecto. Excelente plataforma para trabajar en proyectos de software libre. Aunque también ofrece la posibilidad de crear proyectos privados previo pago o con cuentas especiales para fines educativos.

En el siguiente enlace se puede encontrar el código de la aplicación:

<https://github.com/mapreal19/AgileRoadMap->

Estadísticas

Se han publicado más de 400 *commits* durante el desarrollo de la aplicación. El desarrollo comenzó a finales de Octubre de 2013. Se puede observar en la figura 28 que fue durante Febrero-Marzo cuando se realizó gran parte del trabajo.



Figura 28. Github commits

En la figura 29 se puede observar a qué hora y qué día se realizaban el mayor número de commits. Siendo el martes a las 7pm (17 commits) y el viernes a las 12pm (16 commits) los intervalos con mayor actividad.

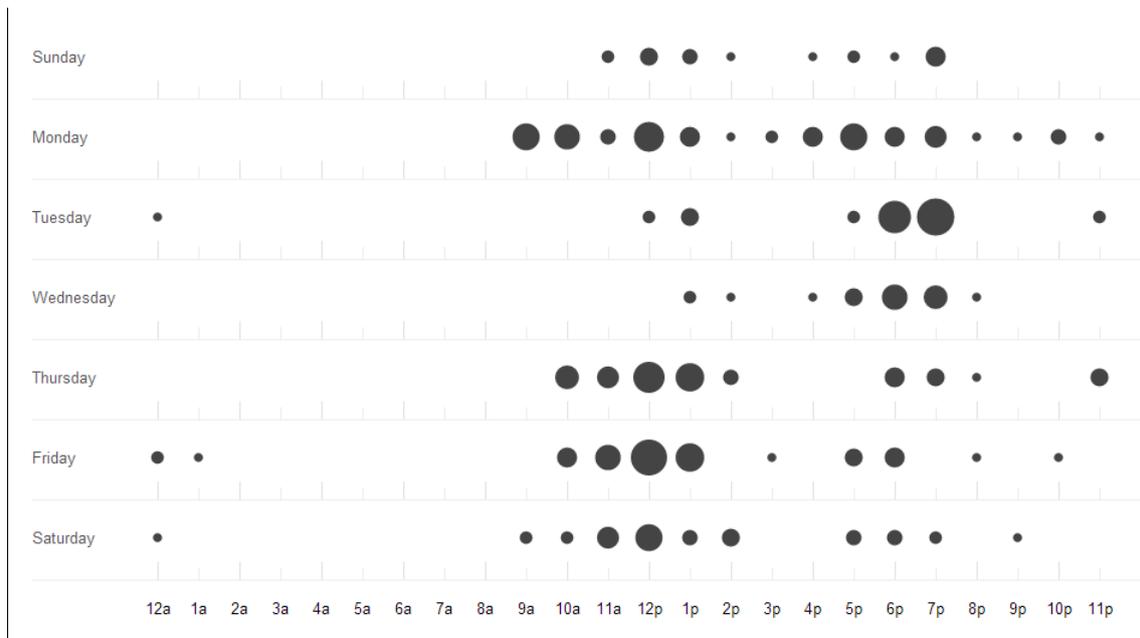


Figura 29. Github frecuencia commits

Branching Model

Se ha seguido el siguiente modelo a la hora del desarrollo de la aplicación [30]:

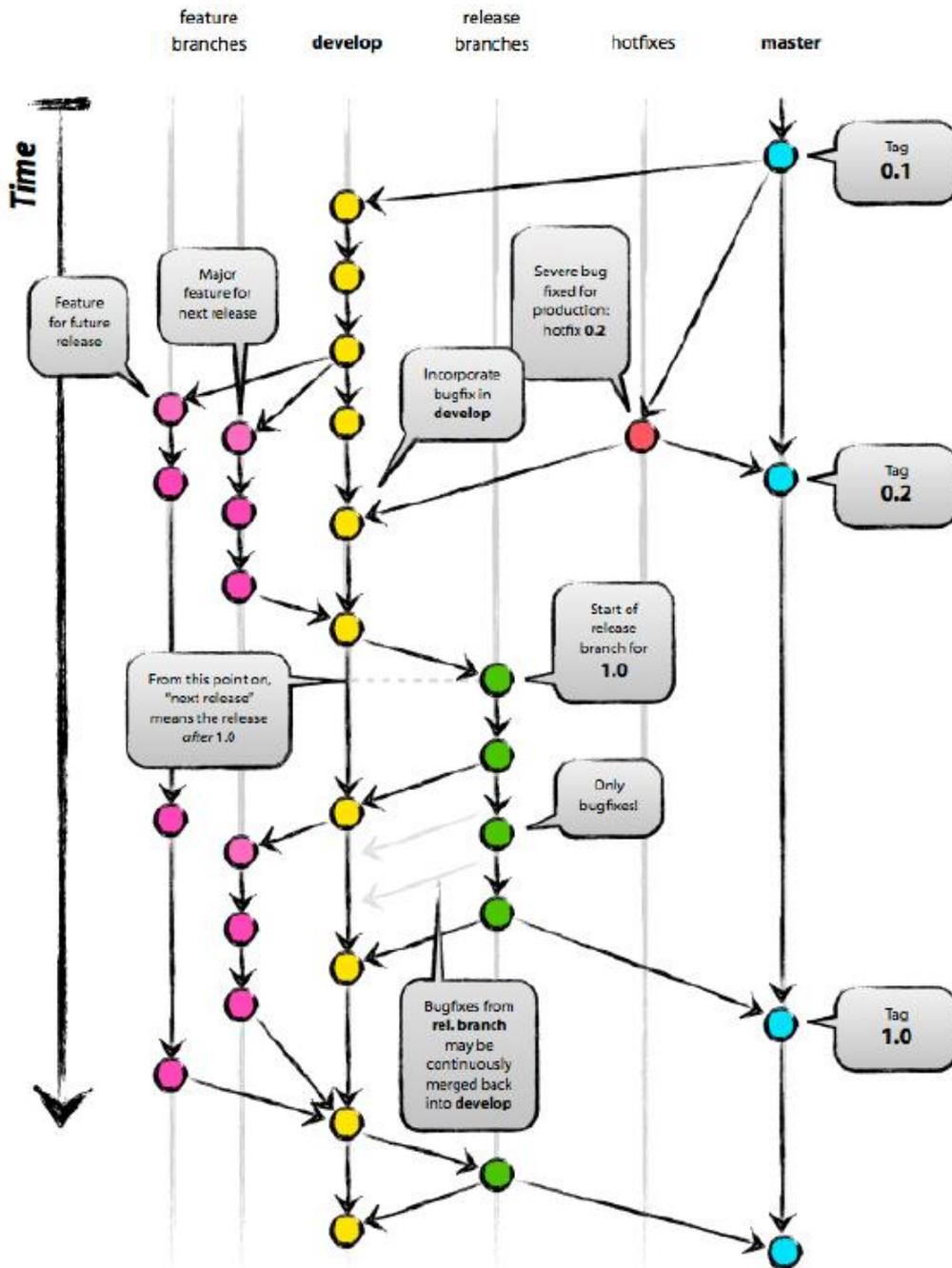


Figura 30. Git Branching Model

Ramas Principales

El repositorio central está compuesto de dos ramas principales con un tiempo de vida infinito:

- master
- develop

La rama *master* representa el código de la aplicación en un estado idéntico al del código en producción.

La rama *develop* refleja un estado con las últimas funcionalidades entregadas para la próxima entrega. Cuando este estado está listo para la puesta en producción, deberá unirse a la rama *master* con un tag representando la versión de la entrega.

Ramas de Apoyo

Son ramas para asistir al desarrollo de la aplicación, facilitar el seguimiento de las funcionalidades, preparación para próximas entregas y para la corrección de forma rápida fallos en producción.

Los tipos son:

- Feature branches
- Release branches
- Hotfix branches

Feature branches

Son usadas para el desarrollo de nuevas funcionalidades que se incorporarán en una entrega futura. Serán incorporadas a la rama *master* una vez completadas, o descartadas si finalmente el resultado no es el esperado.

Release branches

Ramas en las que se preparará la aplicación para su puesta en producción, y en la que se le asignará el número de versión apropiado.

Hotfix branches

Ramas de comportamiento similar a las *release branches*, con la diferencia de que no han sido planeadas. Esto puede ocurrir en el caso de que se encuentre un error en producción en el que se haya de actuar inmediatamente.



<https://www.heroku.com>

El contenido de este apartado está extraído de [31].

Para realizar pruebas de desarrollo utilizaremos los servicios de Heroku. Heroku es uno de los mejores PaaS existentes hoy en día, sus servicios permiten a los desarrolladores utilizar el 100% de su tiempo en la lógica de la aplicación, sin tener que preocuparse por temas de servidores, despliegue de la aplicación o escalabilidad.

Heroku ofrece soporte para Rails y Git, de forma que la publicación/actualización de la aplicación es realmente sencilla:

- `heroku create`: Comando necesario para crear la aplicación.
- `git push heroku <rama>:master`: Actualiza la aplicación.
- `heroku run rake db:migrate`: Ejecuta las migraciones pendientes.

En Heroku una aplicación está compuesta por un número determinado de **Dynos**. Un dyno es un ligero contenedor que ejecuta un único comando especificado por el usuario (ver figura 31). Puede ejecutar cualquier comando disponible en su entorno combinado con el *slug* (copias de la aplicación optimizadas para ser distribuidas entre los dynos) de tu aplicación.

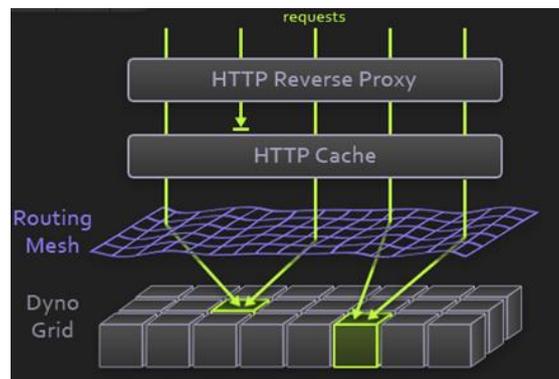


Figura 31. Heroku Dynos

Entre sus características encontramos:

- Elasticidad y escalabilidad: Los números de dynos pueden ser modificados en cualquier momento.
- Encaminamiento: Los routers rastrean la localización de todos los dynos y encaminan tráfico HTTP hacia ellos correspondientemente.
- Gestor de dynos: Dynos en ejecución son monitoreados para asegurar que siguen ejecutándose. Si algún dyno falla otro es lanzado para reemplazarlo.
- Distribución: Dynos son distribuidos de forma que están en localización física diferente.
- Aislamiento: Cada dyno está aislado en su propio contenedor aportando varios beneficios para la seguridad, recursos y robustez de la aplicación.



<https://www.digitalocean.com/>

Hosting que se ha utilizado para la puesta en producción de la aplicación. Entre sus características cabe destacar:

- Precio: Por apenas unos 5\$/mes se nos ha proporcionado un servidor con 512MB de memoria, un procesador, 20GB de espacio en disco SSD, 1TB de transferencia.
- Rapidez: Gracias en parte al uso de unidades de estado sólido.
- Servidor en Europa.
- Facilidad del despliegue de la aplicación.
- 99,99% de tiempo de actividad.

Dokku

DigitalOcean ofrece la instalación de Dokku mediante un simple click. Dokku viene a ser un “mini-Heroku” escrito en unas 100 líneas de *Bash* (<https://github.com/progrium/dokku>), con lo que la aplicación se podrá desplegar de forma idéntica a como se hacía en Heroku sin realizar ningún cambio.

Gracias al tutorial ofrecido por DigitalOcean se pudo configurar la aplicación para correr en Dokku sin ningún problema. Tutorial:

<https://www.digitalocean.com/community/articles/how-to-use-the-dokku-one-click-digitalocean-image-to-run-a-ruby-on-rails-app>

Lo único a tener en cuenta para que la aplicación funcione en el dominio raíz en vez de en un subdomino (ver [32]) se deberá añadir la rama remota de la siguiente forma:

```
$ git remote add dokku dokku@188.226.146.239:agile-roadmap.tuneupprocess.com
```

Comandos útiles

La aplicación tendrá como nombre “agile-roadmap.tuneupprocess.com”. Una vez nos hayamos conectado mediante *ssh* a nuestro servidor se podrán ejecutar los siguientes comandos:

- **dokku help**: muestra todos los comandos disponibles.
- **dokku config <app>**: muestra todas las variables de entorno de la aplicación.
- **dokku config <app> KEY1=VALUE1 [KEY2=VALUE2 ...]**: asigna una o más variables de entorno a la aplicación. Fue necesario asignar a la variable SMTP_PASSWORD la contraseña correcta para el funcionamiento del servidor smtp.
- **dokku logs <app> [-t]**: muestra los últimos logs de la aplicación.
- **dokku postgresql:info <db>**: muestra la información necesaria para conectarse a la base de datos (la base de datos tiene el mismo nombre que el de la aplicación).
- **dokku run <app> <cmd>**: ejecuta un comando en el entorno de la aplicación.

Herramientas y Recursos Utilizados

Para el desarrollo de la aplicación web se hizo uso de las siguientes utilidades.



Sublime Text 3

<http://www.sublimetext.com/3>

Sofisticado editor de textos muy popular entre los desarrolladores. Soporta prácticamente cualquier lenguaje, es multiplataforma y contiene multitud de funcionalidades muy útiles para el programador.



RubyMine

<http://www.jetbrains.com/ruby/>

IDE más popular para el desarrollo en Rails, si no el mejor, para el desarrollo en Rails. Ofrece todas las ventajas que un IDE puede proporcionar. Como punto en contra es que su uso es de pago, aunque se puede disponer de una versión de prueba de 30 días. Si se utiliza para proyectos de software libre su uso es gratuito.



<https://www.nitrous.io/>

Plataforma web donde se puede desarrollar el código de tu aplicación en la nube. Interesante opción si se quiere desarrollar desde un entorno donde no se desee o no se pueda instalar la tecnología necesaria (Ruby, Rails, SQLite, etc.). Dispone también de aplicación de escritorio para Windows y Mac, para aquellos usuarios que quieran editar el código con su editor o IDE preferido (<https://www.nitrous.io/desktop>).



Inspector de Google Chrome

Navegador web desarrollado por Google que se ha utilizado para realizar pruebas durante la implementación. Se hizo uso de su inspector y de RailsPanel (https://github.com/dejan/rails_panel), extensión para Chrome que proporciona gran cantidad de información relativa a Rails (controlador y acción ejecutada, consultas a la base de datos, etc.).

Git Shell

<https://windows.github.com/>

Herramienta desarrollada por GitHub que integra Git en la consola de Windows (usando Powershell). En ella se podrán realizar los comandos necesarios para el control de versiones.



SQLite Database Browser

<http://sqlitebrowser.org/>

Herramienta de funcionalidad simple que te permite realizar consultas SQL sobre una base de datos SQLite, ya que ésta es la que se utiliza en el entorno de desarrollo.

Putty

<http://www.putty.org/>

Cliente SSH para la plataforma Windows. Utilizado para realizar la conexión con el servidor de la aplicación.



<http://railscasts.com/>

Excelente plataforma creada por Ryan Bates con gran cantidad de video tutoriales de gran ayuda para el desarrollo en Ruby on Rails.



<http://stackoverflow.com/>

Plataforma en la que gran cantidad de preguntas relativas a la programación son resueltas por los usuarios. Varios de los desafíos/errores presentes en la aplicación fueron resueltos gracias a esta plataforma.



<https://www.ruby-toolbox.com/>

Notoria plataforma para encontrar todo tipo de librerías escritas en código Ruby. Algunas de las librerías usadas en la aplicación fueron encontradas gracias a esta plataforma.

7. Guía de uso

En la página principal del sitio Agile Roadmap+ se dispone de un vídeo (ver figura 32) en que sirve de ayuda al usuario para entender el funcionamiento de la aplicación web.

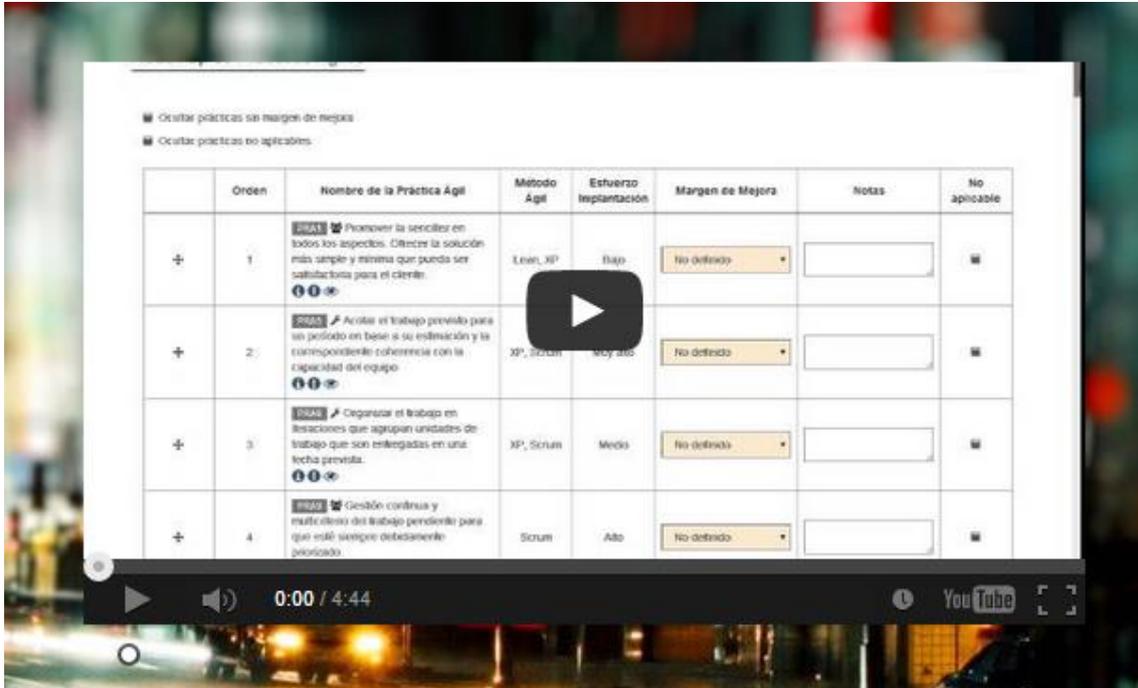


Figura 32. Vídeo ayuda Agile Roadmap+

Además en <http://agile-roadmap.tuneupprocess.com/es/agile-roadmap/pasos-para-elaborar-un-agile-roadmap> se resumen los pasos para elaborar tu propio roadmap de prácticas ágiles. Estos se detallan a continuación:

Un Agile Roadmap+ debería ser el resultado de un diagnóstico y evaluación del contexto del equipo de trabajo. En nuestra experiencia en implantación de prácticas ágiles hemos ido refinando un protocolo para este trabajo de diagnóstico y evaluación, el cual consta de los siguientes pasos:

1. Seleccionar el equipo de trabajo en el cual se aplicarán las prácticas ágiles. Nuestra aproximación al respecto es siempre bottom-up, es decir, implantar prácticas ágiles trabajando muy estrechamente con cada equipo. Si bien es necesario hacer un trabajo previo de promoción hasta un determinado nivel directivo (para conseguir el apoyo de la iniciativa), el trabajo de implantación se hace con el equipo, no debe ser impuesto desde "arriba" ni debe dejarse a la motivación y esfuerzo heroico de algunos integrantes del equipo.
2. Estudiar los productos, servicios y/o proyectos encargados al equipo seleccionado. Obviamente lo ideal sería que el equipo sólo trabajase en un producto, servicio o proyecto, pero desgraciadamente suele ocurrir que el equipo es responsable de varios trabajos diferentes.

3. Evaluar globalmente el tipo de trabajo encargado al equipo y su posible diversidad. Hay dos factores determinantes: (a) si se trata de un producto (desarrollo y/o mantenimiento del producto) o de un servicio, y (b) cómo se organiza el trabajo, si se quiere (y puede) planificar el trabajo o si lo que interesa es atender el trabajo en la medida que se recibe. Respecto de (a), si el trabajo del equipo no está asociado a un producto simplemente se deberían descartar ciertas prácticas que son claramente orientadas a trabajo con productos. En el caso (b), dependiendo de si el trabajo se planifica o si se atiende según se recibe, correspondientemente hay ciertas prácticas aplicables a uno o a otro caso. Para un mismo equipo puede haber mezclas de situaciones respecto de (a) o (b), por ejemplo, el equipo se encarga del desarrollo de cierto producto pero también es responsable de unos servicios, o si además, para unos casos de producto o servicio interesa planificar y para otros no. Es muy importante que para cada caso se adapte adecuadamente el proceso, es decir, que se apliquen las prácticas más adecuadas. Por ejemplo, sería un error que todo el trabajo del equipo se estime y planifique siendo que, el equipo está encargado de un servicio cuya demanda no es previsible y no se quiere (o no se puede) agrupar ni planificar.
4. Establecer los objetivos que pretende la iniciativa de mejorar de proceso. Este paso obliga a una reflexión y diagnóstico respecto del desempeño del equipo en contexto del producto/servicio estudiado. Cada práctica ágil contribuye en cierta medida a ciertos objetivos con lo cual, ordenando por importancia de los objetivos en el contexto del equipo de trabajo se podrán correspondientemente seleccionar prácticas candidatas según su contribución a dichos objetivos.
5. Acotar las prácticas candidatas. Las prácticas candidatas estarán acotadas por la evaluación realizada en el paso 3, es decir, según los factores (a) y (b) algunas prácticas candidatas podrían ser descartadas para los pasos siguientes.
6. Establecer el nivel de aplicación de cada práctica. Es importante evaluar si una práctica está ya aplicada, no aplicada, parcialmente aplicada o simplemente no interesará aplicarla. Una práctica no aplicada tendrá un mayor efecto que una práctica que se está ya aplicando de forma parcial.
7. Considerar el nivel de dificultad que tendrían los desafíos de implantación de cada práctica. Las prácticas no aplicadas o parcialmente aplicadas ofrecen un margen de mejora si se implantan, sin embargo, hay que tener presente los desafíos que normalmente deben enfrentarse al implantar cada práctica.
8. Con toda la información recopilada en los pasos anteriores valorar la prioridad de cada práctica estableciendo un orden de aplicación (Roadmap). Seguir las siguientes directrices: valorar positivamente la importancia de los objetivos a los que contribuye la práctica y valorar también la contribución de cada práctica a los objetivos, valorar positivamente el nivel de aplicación actual de la práctica (mayor si la práctica no está siendo aplicada), valorar el nivel de dificultad de los desafíos de la práctica (mientras menos mejor), valorar el esfuerzo de aplicación de la práctica (mientras menos mejor).

Agile Roadmap+ permite realizar un roadmap de forma muy flexible, ofreciendo además un espacio para el usuario en el cual se almacena su roadmap, de manera que puede acceder a él y completarlo/modificarlo en cualquier momento.

8. Estadísticas Agile Roadmap+

La aplicación fue publicada a fecha de 25 de Abril del año 2014. A partir de entonces se recogió la información proporcionada por los usuarios (ámbito de trabajo, miembros equipo, ordenamiento prácticas, etc.) y se hizo uso de Google Analytics para un análisis más profundo. Con motivo de la redacción de esta tesis, se recogieron los datos a fecha de 10 de Julio del 2014.



“Google Analytics es un servicio gratuito de estadísticas de sitios web. [...] Se pueden obtener informes como el seguimiento de usuarios exclusivos, el rendimiento del segmento de usuarios, los resultados de la campaña de marketing, el marketing de motores de búsqueda, las pruebas de versión de anuncios, el rendimiento del contenido, el análisis de navegación, los objetivos y proceso de redireccionamiento o los parámetros de diseño web”. [33]

Es un servicio de gran utilidad que fue muy fácil de configurar. Gracias a éste se pudo comprobar entre otras cosas qué navegador fue el más usado por los usuarios, y qué sistema operativo móvil utilizaban los usuarios que accedían mediante sus *smartphones*. Esto te permite conocer que sistemas operativos y navegadores se han de tener más en cuenta.

Algunas de las estadísticas más significativas obtenidas de Google Analytics son las siguientes.

Sesiones



Figura 33. Sesiones Analytics

“Una sesión es el periodo durante el cual un usuario interactúa con su sitio web, aplicación, etc.”

Se han obtenido un total de 386 sesiones realizadas por 190 usuarios distintos. Destacar que la duración media de la sesión fue de 3 minutos y 13 segundos. En la figura 33 se ilustra esta información.

País

País/territorio	Sesiones	% Sesiones
1.  Spain	282	37,75%
2.  Paraguay	176	23,56%
3.  Chile	76	10,17%
4.  Argentina	65	8,70%
5.  Colombia	38	5,09%
6.  United States	36	4,82%
7.  Peru	31	4,15%
8.  Ecuador	11	1,47%
9.  Germany	6	0,80%
10.  United Kingdom	4	0,54%

Figura 34. Analytics -- Sesión/País

Como se ilustra en la figura 34 la mayor parte de las sesiones se han producido en los países de España, Paraguay y Chile. El alto número de sesiones en Paraguay puede verse debido a que mi tutor, Patricio Letelier, impartió un curso en dicho país demostrando el uso de esta aplicación.

Navegador

Navegador	Sesiones	% Sesiones
1. Chrome	453	60,64%
2. Firefox	180	24,10%
3. Safari	54	7,23%
4. Internet Explorer	23	3,08%
5. Safari (in-app)	13	1,74%
6. Android Browser	12	1,61%
7. Opera	7	0,94%
8. Mozilla Compatible Agent	3	0,40%
9. BlackBerry	1	0,13%
10. Opera Mini	1	0,13%

Figura 35. Navegadores Analytics

En la figura anterior se puede observar que la mayor parte de los usuarios (74%) accedieron a la aplicación a través de los navegadores *Google Chrome* o *Mozilla Firefox*, por lo que es crítico asegurar el correcto funcionamiento de la aplicación en ambos. No interesaría en este caso invertir mucho tiempo para la correcta experiencia de usuario en el navegador *Opera*.

Sistema Operativo Móvil

Sistema operativo	Sesiones	% Sesiones
1. Android	64	50,79%
2. iOS	61	48,41%
3. BlackBerry	1	0,79%

Figura 36. SO móvil Analytics

En la figura 36 se ilustra que cantidades similares de usuarios accedieron a la aplicación web a través de un dispositivo móvil *Android* o *Iphone*, por lo que en caso de desarrollar una aplicación móvil sería interesante ofrecer una aplicación para ambas plataformas. No interesaría *Windows Phone*.

Flujo del comportamiento

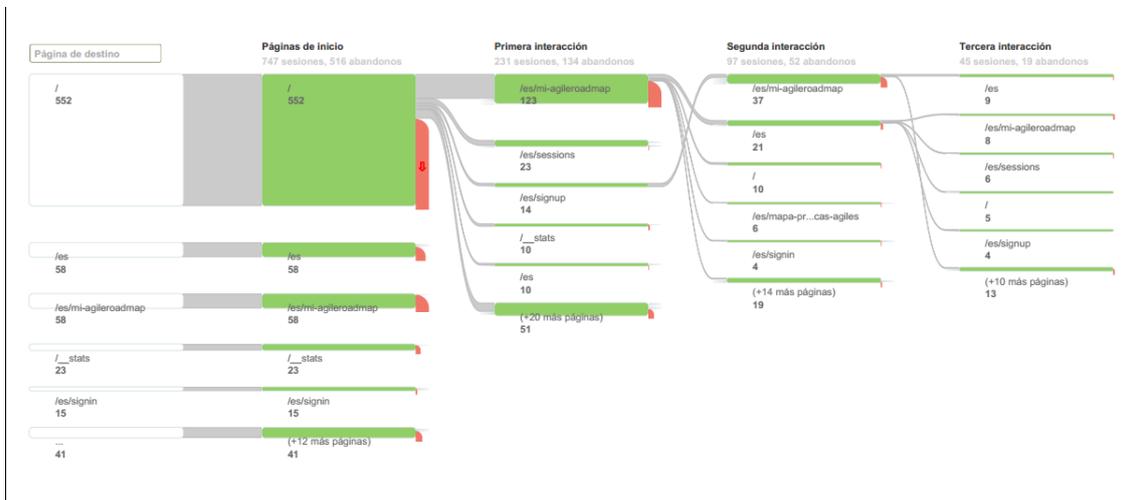


Figura 37. Flujo del comportamiento Analytics

En la figura anterior se puede observar que cuando los usuarios accedían a la página principal de la aplicación, una gran mayoría de ellos (66.3%) abandonaba el sitio sin realizar ningún paso adicional.

Sitio Agile Roadmap+

Con la ayuda de la librería Chartkick (<http://chartkick.com/>) se pudieron obtener gráficas descriptivas a partir del contenido de la base de datos de la aplicación. Estas estadísticas podrán contemplarse en todo momento en tiempo real en el siguiente enlace:

<http://agileroadmap.herokuapp.com/es/agile-roadmap/stats>

A continuación se comentaran cada una de las gráficas.

Usuarios

Se obtuvieron alrededor de 100 usuarios registrados en la aplicación.

Ámbito de Trabajo

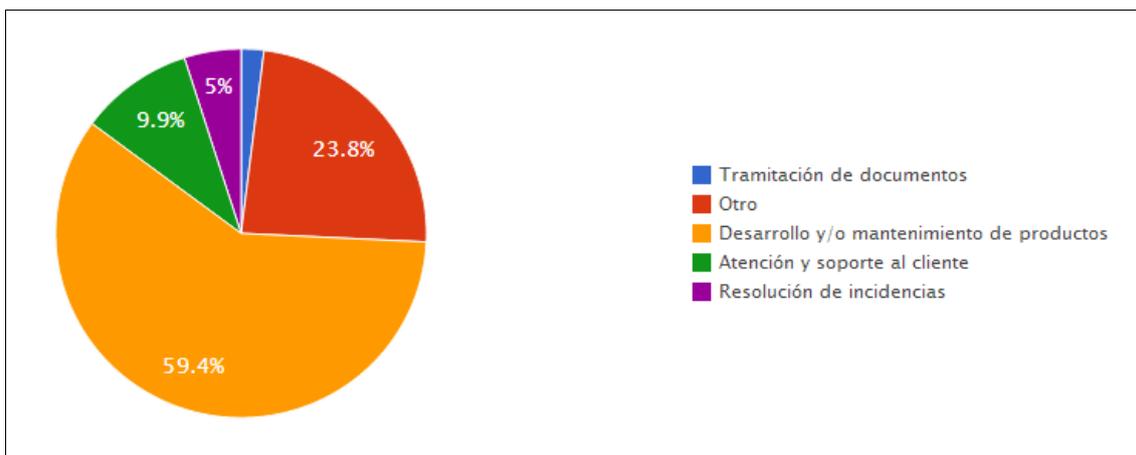


Figura 38. Ámbito de trabajo

Como se observa en la figura 38 más de la mitad de los usuarios (59%) se dedican en su trabajo al desarrollo y/o mantenimiento de productos.

Sector de Empresa

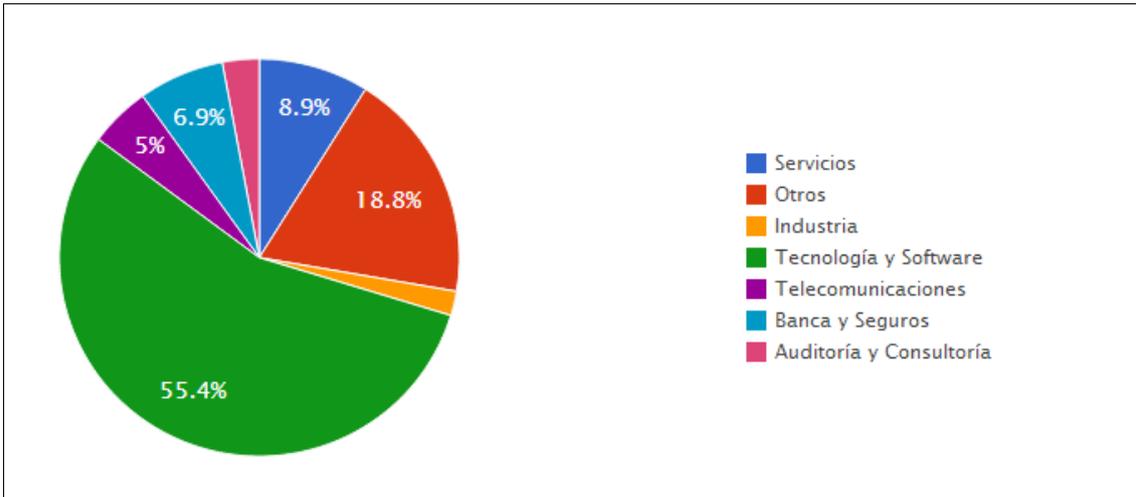


Figura 39. Sector de empresa

En la figura 39 se observa que el sector de empresa mayoritario fue el de Tecnología y Software con un 55.4%.

Número de integrantes de equipo

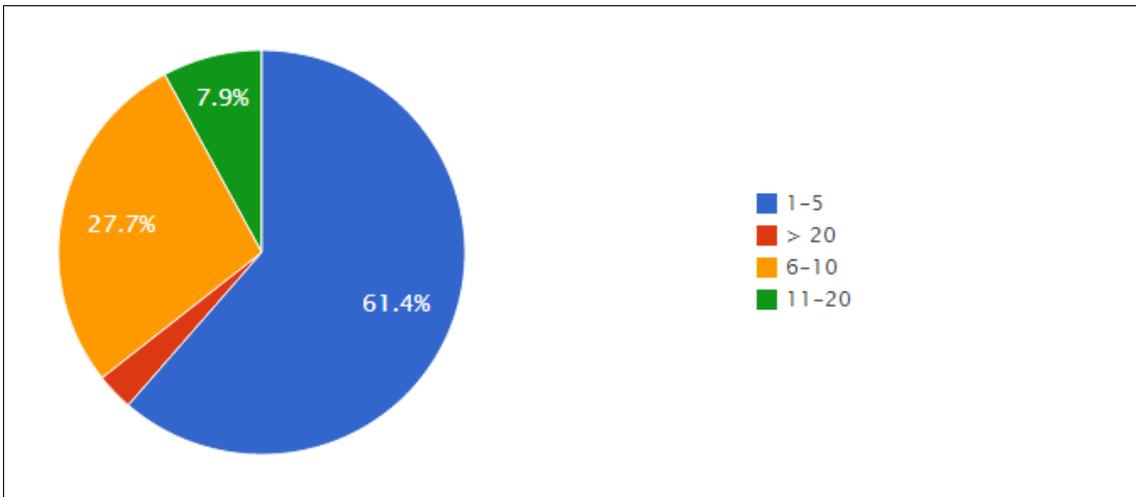


Figura 40. Número de trabajadores

En la figura 40 se ilustra que la mayoría de los usuarios pertenecían a empresas pequeñas con un número de trabajadores reducido (de 1 a 5).

País



Figura 41. Países con usuarios registrados

En la figura 41 se observa que la gran mayoría de los usuarios registrados se dieron en los países de España (38) y Paraguay (43).

Posición media de cada práctica

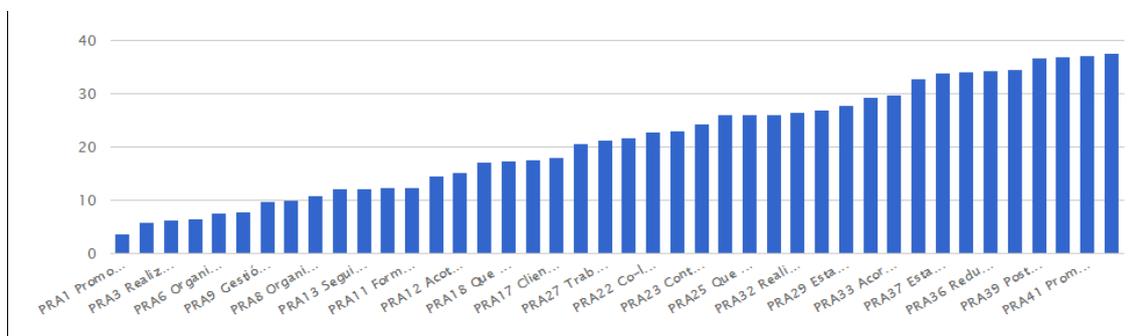


Figura 42. Posición Prácticas Ágiles

En la figura anterior se puede visualizar la posición media de entre todos los roadmaps de los usuarios registrados. Las prácticas que se encontraban más arriba en el roadmap son las siguientes:

- PRA1: Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente. Con una posición media de 3,60.
- PRA2: Abordar y entregar trabajo terminado de forma incremental. Con una posición media de 5,78.
- PRA3: Realizar entregas frecuentes de unidades de trabajo terminadas. Con una posición media de 6,38.

Nivel medio de aplicación de cada práctica

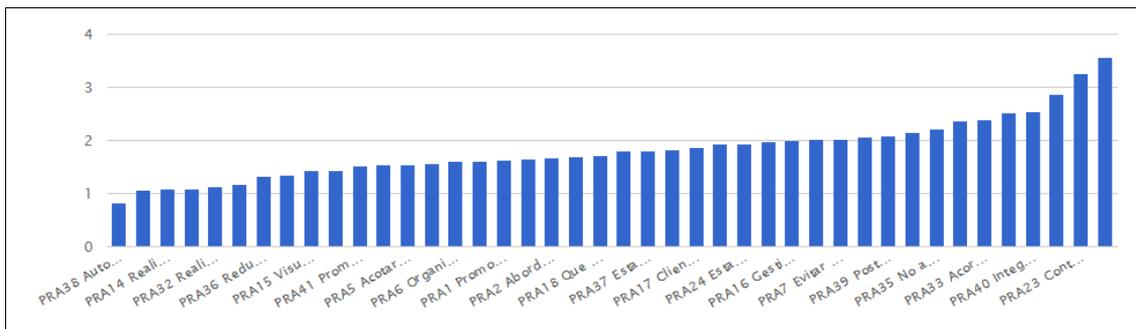


Figura 43. Nivel medio de aplicación de cada práctica

En la figura anterior (ver Figura 43) se pueden observar tanto las prácticas más aplicadas como las no aplicadas. Entre las menos aplicadas destacan:

- PRA38: Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios. Con un valor de 0,82.
- PRA42: Mejorar continuamente la organización interna del producto para facilitar su mantenimiento. Con un valor de 1,07.

Mientras que las más aplicadas son las siguientes:

- PRA22: Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico. Con un valor de 3,57.
- PRA23 Contar con un espacio físico de trabajo que favorezca la interacción entre los miembros del equipo. Con un valor de 3,26.

9. Conclusiones y Trabajo Futuro

Se ha desarrollado una aplicación web para facilitar la implantación de prácticas ágiles en equipos de trabajos y/u organizaciones, no solo orientadas al desarrollo del software. Este sitio proporciona una versión mejorada del antiguo sitio Agile Roadmap. Las principales mejoras son:

- Flexibilidad a la hora de construir el roadmap.
- Mayor usabilidad y mejora de la experiencia de usuario.
- El roadmap se puede guardar en el perfil de usuario para una edición posterior.

En el sitio web se han conseguido alrededor de 100 usuarios registrados, siendo su mayoría provenientes de España y Paraguay. El sector de empresa fue principalmente orientado al desarrollo de productos software.

La aplicación web tuvo un muy buena aceptación por parte de los usuarios en Paraguay, ya que estos usuarios fueron alumnos de un curso impartido por mi tutor Patricio Letelier, en él construyeron un roadmap para equipos en su propia empresa. Además, mi tutor hizo uso de la herramienta en consultoría con varias empresas de forma presencial, siendo el resultado satisfactorio para ambas partes.

Para su implementación se ha hecho uso de tecnologías web de última generación (Ruby on Rails, HTML5, CSS3, jQuery), y con la ayuda de Bootstrap, se han tenido en cuenta aspectos de usabilidad para mejorar el aspecto en dispositivos móviles.

Como nota personal, el uso de estas tecnologías me ha servido para profundizar mis conocimientos en tecnologías web y, aunque he echado en falta trabajar en un equipo compuesto de más programadores, el desarrollo de esta aplicación me ha servido para alcanzar la meta de convertirme en, lo que en inglés se denominaría, un *Full-stack Web Developer*.

Por limitaciones de tiempo han quedado pendientes algunas funcionalidades. Entre ellas:

- **Posibilidad de crear varios roadmap para la implantación de prácticas ágiles.** Sería interesante proporcionar a los usuarios la posibilidad de crear varios roadmap, ya que podría darse el caso de que en una empresa los equipos de trabajo o sus líneas de trabajo presenten diferencias significativas en la aplicación de prácticas ágiles.
- **Internacionalización de la aplicación.** La aplicación soporta actualmente el idioma español. Sería interesante soportar otros idiomas como el inglés. Para ello se deberán localizar todos los textos contenidos en la aplicación y asignarles una clave, para poder realizar el cambio de idioma. Los archivos con los contenidos de clave-valor para un idioma en concreto se encuentran en el directorio *config/locales*.

Además, si se quiere traducir texto contenido en la base de datos, una librería como *Globalize* deberá de ser utilizada (<https://github.com/globalize/globalize>).

- **Establecimiento de una conexión segura a la aplicación.** Garantizar la identidad del servidor mediante SSL (en inglés, *Secure Sockets Layer*). Actualmente la conexión con la aplicación no está cifrada, lo que podría ocasionar el robo de la contraseña proporcionada por un usuario al realizar el *Login*. En la aplicación no existe información del usuario de importancia, pero son muchos los usuarios que usan la misma contraseña para varios sitios web, por lo que se ha de evitar que ésta sea descubierta por agentes externos.
- **Testing automatizado.** Sería interesante proporcionar una suite de tests a alto nivel que comprueben el correcto funcionamiento de las principales funcionalidades de la aplicación, para evitarse así el *testing* manual. En Rails se puede conseguir gracias a Capybara (<https://github.com/jnicklas/capybara>), framework para construir tests de aceptación, en donde se simula cómo un usuario interacciona con la aplicación. El tener una buena suite de tests facilita en gran medida la refactorización del código y te asegura que funcionalidades de la aplicación no dejan de funcionar al crear otras nuevas.
- **Panel de administración.** Creación de una interfaz de administración en donde poder visualizar y editar la información relativa a la aplicación (usuarios, objetivos, prácticas ágiles, etc.). ActiveAdmin es el framework más interesante para realizar esta tarea (<http://activeadmin.info/>).

Referencias

- [1] P. Letelier, «Curso Gestión Ágil de Proyectos,» 2014.
- [2] M. Cohn, «Capítulo 1. Why Becoming Agile Is Hard,» de *Succeeding with Agile. Software Development Using Scrum*, 2009.
- [3] G. Heileman, «Coursera - Web Application Architectures (Module 1, Lecture 2),» 2014. <https://www.coursera.org/course/webapplications>.
- [4] 2001. <http://agilemanifesto.org/>. [Último acceso: 2014].
- [5] S. Ratking, «Manifiesto Elicits Cynicism,» *IEEE Computer*, 2001.
- [6] «edX - CS69.1x Software as a Service,» [En línea]. Available: <https://www.edx.org/course/uc-berkeleyx/uc-berkeleyx-cs169-1x-engineering-1377>.
- [7] Versionone, «State of agile survey,» 2014.
- [8] J. Boeg, *Priming Kanban*, 2011.
- [9] T. P. Mary Poppendieck, *Lean Software Development: An Agile Toolkit*, 2003.
- [10] K. Beck, *Extreme Programming Explained: Embrace Change*, 1999.
- [11] L. J. N. Rising, «The Scrum Software Development Process for Small Teams,» 2000.
- [12] M. Cohn, «Chapter 2. Adapting to Scrum,» de *Succeeding with Agile. Software Development Using Scrum*, 2009.
- [13] M. Cohn, «Chapter 4. Iterating Toward Agility,» de *Succeeding with Agile. Software Development Using Scrum*, 2009.
- [14] P. L. Fausto I. Nelson Amancio, «AGILE Roadmap: diagnóstico y evaluación de prácticas ágiles para ser implementadas en equipos de trabajo,» 2013.
- [15] M. Cohn, «Determining how agile you are comparatively,» 2014. [En línea]. Available: <http://www.mountangoatsoftware.com/articles/determining-how-agile-you-are-comparatively>. [Último acceso: 2014].
- [16] S. Ambler, «The Agile Maturity Model (AMM),» <http://www.drdoobbs.com/architecture-and-design/the-agile-maturity-model->

- amm/224201005. [Último acceso: 2014].
- [17] «Enterprise Agility Maturity Matrix,» 2013. <http://blogs.atlassian.com/2013/11/enterprise-agility-maturity-matrix/>.
- [18] P. Letelier, «Agile Roadmap: el primer paso para la implantación de prácticas ágiles,» 2013. <http://agilismoatwork.blogspot.com.es/2013/04/agile-roadmap-el-primer-paso-para-la.html>. [Último acceso: 2014].
- [19] P. Letelier, «Carta de prácticas ágiles,» 2013. <http://agilismoatwork.blogspot.com.es/2013/02/carta-de-practic-as-agiles-armatu.html>. [Último acceso: 2014].
- [20] «The Ruby Programming Language,» 2014. <http://www.informit.com/articles/article.aspx?p=18225>.
- [21] «CoffeeScript,»: <https://github.com/jashkenas/coffeescript>.
- [22] «jQuery.com,» <http://jquery.com/>.
- [23] «What is Rails? - RailsGuides,» 2014. http://guides.rubyonrails.org/getting_started.html#what-is-rails.
- [24] «What is Rails? - RailsGuides,» http://guides.rubyonrails.org/getting_started.html#what-is-rails. [Último acceso: 16 2013].
- [25] M. Fowler, Patterns of enterprise application architecture, 2003.
- [26] «The Rails Command Line,» 2013. http://guides.rubyonrails.org/command_line.html. [Último acceso: September 2013].
- [27] «Twitter Bootstrap - Wikipedia,» 2014. <http://getbootstrap.com/>.
- [28] «WalkerSands Mobile Traffic Report Q3-2013,» 2013. <http://www.walkersandsdigital.com/Walker-Sands-Mobile-Traffic-Report-Q3-2013>. [Último acceso: 2014].
- [29] C. Duan, «Understanding Git,» 2013. <http://www.sbf5.com/~cduan/technical/git/git-1.shtml>. [Último acceso: 26 2013].
- [30] «A Successful Git Branching Model,» 2014. <http://nvie.com/posts/a-successful-git-branching-model/>.
- [31] «Dynos and the Dyno Manager,» 2014.

<https://devcenter.heroku.com/articles/dynos>.

[32] «Dokku - Deploying to root and subdomains,» 2014. <https://github.com/progrium/dokku/wiki/Deploying-to-root-and-subdomains>.

[33] «Google Analytics,» 2014. http://www.google.com/intl/es_ALL/analytics/. [Último acceso: 2014].