



UNIVERSIDADE DA CORUÑA

FACULTADE DE INFORMÁTICA

DEPARTAMENTO DE TECNOLOXÍAS DA INFORMACIÓN E DAS COMUNICACIÓNS

PROXECTO DE FIN DE CARRERA
ENXEÑERÍA INFORMÁTICA

Aplicación web e móvil para a xestión electrónica de actas deportivas.

Autor/a: Pablo Castro Valiño
Director/a: Santiago Saavedra López
Tutor/a: Fernando Bellas Permuy

A Coruña, a 20 de xuño de 2016.

Agradecementos

Quero aproveitar este oco para recordar a tódolos que me acompañástedes nesta aventura, a tódolos que dende o primeiro día estivéstedes empurrando tamén para que este proxecto saíse adiante.

En primeiro lugar quero comenzar por Jose Manuel da Federación de Peñas que tanto lle dei a vara e tantas horas invertiu de xeito totalmente desinteresado en axudarnos.

A Carlota, David e a tódolos emprendedores que coñecín ao longo destes anos polos ánimos que me infundiron e ese apoio que sempre atopei en eles.

Por suposto a Fernando Bellas como o meu titor e a Universidade da Coruña pola enorme labor docente que me facilitou dispor dos coñecementos para sacar adiante este proxecto.

Pero os apoios máis importantes son aqueles que están contigo día tras día e que nunca flaquean, aqueles que te empurran incondicionalmente como os que meus pais Manolita e Jose, meu irmán Adrian, miña avoa Josefa, meus tíos e toda a familia me deron durante tantos anos.

Non vou olvidar os enormes momentos que pasei na facultade con tódolos meus compañeiros e en concreto quero agradecerlle a Santi toda a súa axuda neste proxecto e as inoxidables aventuras que vivimos xuntos nos últimos anos.

A Ali, que é a culpable principal de que conseguixe rematar este proxecto, por todas esas horas trasnoitando ao meu lado, empurrándome, tranquilizándome e conseguindo sacarme unha sonrisa nos peores momentos, sin ela nada sería o mesmo.

Aos meus amigos, Revi, Alex, Gato, Javi, Chava e Duda que tantas vivencias pasamos xuntos dende moi pequeninos, por esas tardes na biblioteca de Intercentros ou esas noites de videoxogos na residencia de estudiantes.

E por último quero darlle as gracias a GPUL por todo o que contribuiu a miña formación persoal e profesional, a tódolos valores que aprendín aquí e a tódalas oportunidades que me está a brindar.

A todos vós, ¡moitas gracias!

Pablo Castro Valiño
A Coruña, 20 de xuño de 2016

Resumo

Pese aos avances nas Tecnoloxías da Información e das Comunicacións, e na Enxeñería do Software, a xestión de competicións deportivas continúa a atoparse extremadamente atrasada tecnolóxicamente e as federacións e asociacións deportivas invirten gran cantidade de tempo en centos de trámites que teñen que facer de forma manual, e entre os que se atopa a redacción, distribución, revisión e finalmente publicación das actas cos datos estadísticos dos encontros da súas competicións.

Con este proxecto intodúcese unha aplicación móvil para que os árbitros poidan cubrir ditas actas directamente no seu teléfono, permitindo manter actualizados os resultados e as estadísticas dos mesmos en tempo real e mesmo traballar de forma offline.

Actualmente dende a iniciativa VACmatch, estamos impulsando un sistema de xestión de competicións co fin de darlle aos xestores de federacións unha ferramenta na que realizar o seu traballo diario de forma electrónica. Este proxecto, VACmatch Mobile, intégrase nesta ferramenta.

Decidíuse empregar tecnoloxías web na implementación deste desenvolvemento co fin de facilitar a súa utilización en calquera plataforma móvil ou web así como pola versatilidade que aportan.

Palabras clave:

- ✓ VACmatch
- ✓ Aplicación híbrida.
- ✓ ReactJS.
- ✓ Javascript.
- ✓ PouchDB.
- ✓ CouchDB.
- ✓ Deporte.
- ✓ Gestión de competiciones.
- ✓ Software Libre.
- ✓ Open Source.

Índice xeral

	Página
1. Introdución	1
1.1. O deporte amateur e o avance tecnolóxico	1
1.2. A problemática	2
1.3. VACmatch	2
1.4. Resumo do proxecto	3
1.4.1. VACmatch Mobile	3
1.4.2. VACmatch Web	3
1.4.3. Estrutura da memoria	4
2. Estado da arte	6
2.1. A xestión de competicións e actas deportivas	6
2.2. Competidores no mercado	8
2.2.1. Follas de cálculo	8
2.2.2. Ferramentas tradicionais	9
2.2.3. Ferramentas na nube	11
2.2.4. Táboa comparativa	14
2.2.5. Outras plataformas	14
2.3. Aplicacións libres no mercado	14
2.4. Solución aberta e adaptable	15
3. Metodoloxía	16
3.1. Lean Startup	17
3.2. eXtreme Programming	17
3.3. Scrum	17
3.4. Adaptación da metodoloxía	18

3.4.1. Desenvolvemento orientado ao cliente	18
3.4.2. Sprints con backlog adaptable	18
3.4.3. Reunións semanais	18
3.4.4. Reunións diarias	19
3.4.5. Releases	19
3.4.6. Simplicidade	19
3.4.7. Tests	19
3.4.8. Fluxo de contribución ao proxecto	20
4. Análise de requisitos globais	21
4.1. Consultas a xestores de federacións	21
4.2. Peticións obtidas	22
4.3. Requisitos finais	23
4.3.1. Usuarios	23
4.3.2. Listar actas	23
4.3.3. Visualizar actas	23
4.3.4. Xeración de actas offline	23
4.3.5. Modificación de actas	24
5. Planificación e seguimento	25
5.1. Validación de negocio (Xullo 2015 – Novembro 2015)	26
5.1.1. Prototipo visual	26
5.1.2. MVP funcional	27
5.2. Desenvolvemento de produto (Novembro 2015 – Xaneiro 2016)	29
5.2.1. 1 ^a e 2 ^º iteración. Creación do proxecto e xestión de actas	30
5.2.2. 3 ^a iteración. Eventos	32
5.2.3. 4 ^a iteración. Xestión de usuarios e creación offline de actas	34
5.2.4. 5 ^a iteración. Sinaturas	36
5.3. De empresa a comunidade (Xaneiro 2016 – Maio 2016)	36
5.3.1. 6 ^a e 7 ^a iteración. Optimización e melloras	37
5.3.2. 8 ^a iteración. Testing e integración continua	38
5.3.3. 9 ^a e 10 ^a iteración. Inxección de dependencias	39
5.3.4. Release 0.2.0: Usabilidade en menús	41
5.3.5. Release 0.2.1: I18n e app híbrida	43

5.3.6. Release 0.2.2: Finalización da memoria e revisión de erros	44
6. Fundamentos tecnolóxicos	46
6.1. Linguaxes e frameworks empregados	46
6.2. Bases de datos	47
6.3. Estándares de comunicación	47
6.4. Repositorios de código	48
6.5. Ferramentas de xestión	48
6.6. Ferramentas documentais	49
7. Deseño e implementación	50
7.1. ReactJS e Flux	51
7.1.1. Introdución e elección da tecnoloxía	51
7.1.2. Elementos básicos	52
7.1.3. Estrutura da aplicación	54
7.2. Bases de datos e funcionamento offline	55
7.2.1. PouchDB	56
7.2.2. CouchDB	56
7.2.3. Sincronización e xestión de conflictos	56
7.3. App híbrida con Apache Cordova	56
7.4. Interface gráfica e usabilidade	58
7.4.1. Elementos comúns	58
7.4.2. Iniciar sesión	62
7.4.3. Listado de actas	63
7.4.4. Acta	63
7.4.5. Finalización do encontro	66
7.4.6. Modificación de estilos	68
7.5. Multideporte	69
7.5.1. Deporte	69
7.5.2. Roles de usuarios	70
7.5.3. Eventos	70
7.6. Deseño da DB	72
7.7. I18n	73
7.8. Inxección de dependencias	73

7.9. Testing e Integración continua	74
7.9.1. TDD e BDD	75
7.9.2. Jest	75
7.9.3. Travis CI	78
8. Conclusións e traballo futuro	79
8.1. Recoñecementos	79
8.1.1. Finalista no Certamen de Proyectos Libres da UGR	79
8.1.2. Premio Universitario de Software Libre	80
8.2. Traballo futuro	80
8.2.1. Melloras de desenvolvemento	80
8.2.2. Creación de comunidade	81
8.3. Conclusións	82
A. Configuración e execución da aplicación.	84
A.1. Execución dende plataforma Web.	84
A.2. Execución dende plataforma móvil.	85
B. Glosario de acrónimos	86
C. Glosario de términos	87
Bibliografía	89
GNU Free Documentation License	90
1. APPLICABILITY AND DEFINITIONS	90
2. VERBATIM COPYING	92
3. COPYING IN QUANTITY	92
4. MODIFICATIONS	93
5. COMBINING DOCUMENTS	94
6. COLLECTIONS OF DOCUMENTS	95
7. AGGREGATION WITH INDEPENDENT WORKS	95
8. TRANSLATION	95
9. TERMINATION	96
10. FUTURE REVISIONS OF THIS LICENSE	96
11. RELICENSING	97

ADDENDUM: How to use this License for your documents	97
--	----

Índice de figuras

Figura	Página
2.1. Exemplo de acta deportiva	7
2.2. Folla de cálculo	8
2.3. Aplicación web de Novanet	9
2.4. Web da FGVB co sistema Federatio	10
2.5. App móvil de miLeyenda	11
2.6. Torneo de padel na web de Esportics	12
2.7. App móvil de Sportngin	13
5.1. Web do I Torneo VACmatch	29
5.2. Diagrama de Gant do sprint 1	31
5.3. Diagrama de Gant do sprint 2	32
5.4. I Lonxa de Financiamento Responsable	32
5.5. Diagrama de Gant do sprint 3	34
5.6. Diagrama de Gant do sprint 4	35
5.7. Diagrama de Gant do sprint 5	37
5.8. Diagrama de Gant do sprint 7	38
5.9. Diagrama de Gant do sprint 8	39
5.10. Diagramas de Gant dos sprints 9 e 10	41
5.11. Finalistas CUSL	42
5.12. Diagramas de Gant da release 0.2.0	42
5.13. Diagramas de Gant da release 0.2.1	43
5.14. Diagramas de Gant da release 0.2.2	45
7.1. Esquema da arquitectura Flux	52

7.2. Arquitectura dunha aplicación con Apache Cordova	57
7.3. Menú lateral esquierdo.	58
7.4. Menú desplegable derecho.	59
7.5. Ventás de información e axustes.	60
7.6. Barra de notificacións.	60
7.7. Vista de inicio de sesión.	63
7.8. Listado de actas.	64
7.9. Vista principal dun acta.	65
7.10. Seleccionar xogadores presentes no encontro.	66
7.11. Fin de encontro.	67
7.12. Sinatura dun acta.	68
7.13. Patron estratexia para o deporte.	69
7.14. Diagrama parcial da vista dos eventos.	71
7.15. Diagrama de clases para a xestión de eventos deportivos.	72
7.16. Esquema de base de datos de Person.	72

Índice de cadros

Tabla	Página
2.1. Táboa comparativa das alternativas existentes.	14

Capítulo 1

Introdución

Índice xeral

1.1. O deporte amateur e o avance tecnolóxico	1
1.2. A problemática	2
1.3. VACmatch	2
1.4. Resumo do proxecto	3
1.4.1. VACmatch Mobile	3
1.4.2. VACmatch Web	3
1.4.3. Estrutura da memoria	4

NESTE capítulo trataranse os aspectos básicos para comprender o proxecto así como os motivos que levaron ao seu desenvolvemento e a estrutura da presente memoria.

Falaremos do estado do deporte na actualidade e da xestión deportiva en concreto, co fin de mostrar a necesidade de impulsar un proxecto dentro deste campo para posteriormente introducir a iniciativa dentro da que xurdiu este proxecto e rematando con un resumo da problemática que resolve.

1.1. O deporte amateur e o avance tecnolóxico

Actualmente o deporte é fundamental na vida das persoas, durante os últimos anos o número de españois que realizan algunha actividade física medrou enormemente así como o número de competicións amateur, que permiten a estos deportistas, competir por un custe moito más asequible que as federacións oficiais.

Se embargo, este crecemento do número de deportistas non veu acompañado tamén dunha renovación tecnolólica das competicións polo que gran parte dos seus xestores seguen a invertir un tempo elevado nas súas competicións e non teñen apenas relación dixital coas persoas que compiten nas mesmas.

1.2. A problemática

Actualmente os organizadores de competicións deben realizar unha serie de tarefas que se describen a continuación e que, na súa meirande parte, realizan de forma manual ou axudados de follas de cálculo, ao non dispor das ferramentas tecnolóxicas axeitadas a un prezo accesible.

Inscricóns Na maior parte das competicións, os xogadores seguen a ter que levar cuberta a súa ficha cos seus datos persoais en papel, fotocopia do DNI, fotografía, etc. para que a federación garde eses datos nunha folla de cálculo.

Aprazamentos de partidos Moitas esíxenllas aos equipos, unha vez postos de acordo, enviar unha confirmación en papel, por correo ordinario ou fax.

Notificacións Deben avisar aos sancionados, os cambios no calendario, etc. cando menos por correo electrónico.

Revisión de sancionados A federación debe comprobar que un xogador sancionado non xogou un partido que non debía.

Loxística das actas dos encontros O árbitro do encontro debe recoller as actas na asociación e volver traelas cubertas despois dos encontros.

Publicación de resultados A federación debe recopilar tódolos datos das actas para publicalos, ben sexa nunha web ou por email aos participantes.

Publicación de clasificacións e estadísticas A federación debe calcular a clasificación e recopilar as estadísticas para publicalas posteriormente.

O proxecto desenvolvo trata de resolver os últimos apartados mencionados no punto anterior, *a xestión das actas dos encontros, a súa loxística e a automatización da publicación de resultados e clasificacións*.

Para iso decidíuse crear unha aplicación móvil que permita que os árbitros xestionen as súas actas directamente dende o seu teléfono móvil ou tableta, nunha aplicación multidispositivo baseada en tecnoloxías web, permitindo incluso realizar ditas actas sen conexión a internet, algo que hoxe en día ningunha aplicación ofrece no mercado nacional.

Esta aplicación móvil integrarase tamén con un sistema de xestión de competicións co fin de que os árbitros poidan cubrir as actas e publicar as estadísticas e resultados directamente na web da federación, a través do seu sistema de xestión.

1.3. VACmatch

VACmatch é unha iniciativa empresarial xurdida na Universidade da Coruña para mellorar a xestión de competicións deportivas a través dunha serie de aplicacións entre as que se atopa este proxecto.

A iniciativa recibiu o pasado ano a calificación de *Iniciativa Empresarial de Base Tecnolóxica (IEBT)* recoñecendo o seu grao de innovación así como participou en diversos programas de apoio a ideas emprendedoras como *Yuzz*¹ ou *Telefónica Galicia OpenFuture_*² e durante case un ano, conviviu con outras iniciativas empresariais no *Viveiro de empresas da Universidade da Coruña*.

Ano e medio despois do seu comezo decidiuse abandonar o proxecto como iniciativa empresarial pero VACmatch continúa como comunidade baseada nun proxecto de software libre.

1.4. Resumo do proxecto

O proxecto desenvolvo componse de dúas partes diferenciadas que permiten a xestión das actas dos encontros por parte das federacións deportivas.

1.4.1. VACmatch Mobile

É unha aplicación móvil híbrida realizada con tecnoloxías web co fin de poder utilizala en calquera plataforma, tanto a través da web como nun móvil Android, IOS, FirefoxOS... e que permitirá aos árbitros das competicións realizar tódalas xestións coas actas dos encontros dende o seu teléfono.

Lista de actas Esta aplicación permite que os árbitros poidan dispoñer no seu teléfono das actas dos partidos que teñen que dirixir, coa localización e a data dos mesmos e que se actualizan de forma automática cando se reasigan ou se cancelan.

Convocatoria de xogadores Unha vez o árbitro chega ao encontro pode seleccionar na aplicación os xogadores que asistiron ao mesmo únicamente con un click, introducir a algúin novo se o deseja ou editar datos como o dorsal dun xogador.

Xestión de actas Unha vez comezado o encontro, a aplicación permitirá introducir os diversos eventos que ocorren no mesmo como infraccións, goles ou tarxetas, pensando dende o primeiro momento para que resulte sinxelo engadir novos deportes e eventos se é preciso.

Sinatura de actas Para rematar o encontro, o árbitro poderá engadir comentarios a mesma acta e tanto él como os xogador ou persoal dos equipos, poderán asinar a acta con un código PIN do que dispón cada un.

Actas offline O árbitro poderá crear actas incluso aínda que non tivese sincronizados tódolos datos do partido, permitindo cubrir as actas incluso no peor escenario posible.

1.4.2. VACmatch Web

A aplicación móvil explicada anteriormente atópase nas primeiras fases de un proceso de integración cunha aplicación web, a través da cal as federacións poden xestionar completamente as súas

¹Programa de formación empresarial do Centro Internacional Santander Emprendimiento

²Programa de mentorización e formación de negocio de Telefónica con un premio de 2.000 €

competiciones, modificar o calendario, engadir novos xogadores ou equipos, xestionar arbitraxes, etc.

Actualmente permítese que a federación cree as actas dos encontros, os árbitros as sincronicen nos seus teléfonos e unha vez cubertas, tódolos datos sexan publicados automáticamente na web da federación a través dun pequeno plugin.

Esto facilita que a federación poida dar o primeiro paso de sustituir as actas e fichas físicas por versións dixitais pero deben continuar facendo a integración dos datos das actas de xeito manual, xa que ambos sistemas funcionan de momento por separado, e polo tanto é preciso comprobar as actas e mover os seus datos ao sistema de xestión de VACmatch Web onde se gardan os datos finais verificados pola federación.

Nun futuro cercano permitirase manter as clasificacións actualizadas en todo momento sen apenas intervención humana e aforrando un enorme traballo na revisión das actas.

1.4.3. Estrutura da memoria

A memoria componse dun total de 8 capítulos e 5 apéndices nos que se expoñen os pasos que se deron para crear VACmatch Mobile.

Capítulo 1. Introdución. O capítulo actual e no que se tratar de explicar a problemática que levou ao desenvolvemento e o contexto no que se enmarca así como un pequeno resumo da estrutura da memoria.

Capítulo 2. Estado da arte. Explicarase o estado actual das solucións para xestión de competiciones e actas deportivas co fin de mostrar a necesidade existente dunha solución aberta e adaptable.

Capítulo 3. Metodoloxía. Detallaranse as metodoloxías de desenvolvemento seguidas ao longo do proxecto, un conxunto de prácticas e métodos inspirados en varias metodoloxías.

Capítulo 4. Análise de requisitos globais. Explicarase cómo se levou a cabo a análise de requisitos para esta aplicación e detallarase cada un dos mesmos.

Capítulo 5. Planificación e seguimento. Falarase das múltiples iteracións nas que se dividiu o proxecto e da planificación e seguimento das mesmas.

Capítulo 6. Fundamentos tecnolóxicos. Enuméranse as diversas ferramentas e tecnoloxías utilizadas ao longo do proxecto.

Capítulo 7. Deseño e implementación. Neste capítulo explicaranse múltiples detalles de deseño e implementación, decisións tomadas en estos apartados e o motivo polo que se tomaron.

Capítulo 8. Conclusóns e traballo futuro. Contaranse os diversos recoñecementos recibidos polo proxecto, as liñas de traballo futuro nos apartados de desenvolvemento e creación de comunidade e as conclusóns obtidas finalmente.

Apéndices Contén a información para poder executar a aplicación, varios glosarios, a bibliografía e a licenza escollida para a documentación.

Capítulo 2

Estado da arte

Índice xeral

2.1.	A xestión de competicións e actas deportivas	6
2.2.	Competidores no mercado	8
2.2.1.	Follas de cálculo	8
2.2.2.	Ferramentas tradicionais	9
2.2.3.	Ferramentas na nube	11
2.2.4.	Táboa comparativa	14
2.2.5.	Outras plataformas	14
2.3.	Aplicacións libres no mercado	14
2.4.	Solución aberta e adaptable	15

NESTE capítulo mostrárase cómo as federacións deportivas están xestionando actualmente as súas competicións, cales son as diversas alternativas no mercado para isto e tamén se fará unha análise do software libre neste campo.

Búscase mostrar a necesidade existente de impulsar unha alternativa libre e con boa usabilidade que actualmente está a reclamar o mercado para a xestión de actas electrónicas.

2.1. A xestión de competicións e actas deportivas

A xestión de competicións e eventos deportivos é unha tarefa que dende fai moito tempo se ven realizando de xeito manual e nos últimos anos comezan a aparecer no mercado as primeiras aplicacións para facilitar este traballo.

O proceso comeza cando os xogadores son inscritos a través do seu equipo, nunha competición que xestiona unha federación ou asociación deportiva. Cando os clubes realizan o pago da inscrición, reciben unha ficha identificativa¹ para cada xogador e persoal do mesmo e que deberán levar aos encontros para poder participar.

¹É un documento con fotografía, similar ao DNI e que permite identificar ao integrante do equipo

Unha vez tódolos equipos están inscritos, a federación procede a crear un calendario que contén unha listaxe de enfrontamentos a realizar entre os equipos.

Semanalmente os árbitros da federación son asignados aos partidos que durante esa semana se disputan e deben pasar polo local da mesma para recoller un modelo en papel da acta dos encontros similar ao que se mostra na Figura 2.1 e tamén para coñecer os partidos que terán ao seu cargo, a data e a localización.

O árbitro débese desplazar ao campo ou pista onde se disputa o encontro, recolle as fichas dos xogadores de ambos equipos e copia dentro da acta aqueles que están presentes, descartando os que non asistiron.

Figura 3.1: Exemplo de acte deportiva

Durante o partido, o árbitro ou o segundo árbitro segundo corresponda, encárgase de ir cubrindo todos os eventos que se producen durante o xogo, por exemplo goles, faltas, etc.

Unha vez rematado, os capitáns ou adestradores de ambos equipos deben revisar a acta e asinala se están de acordo co que alí se indica, para que posteriormente o árbitro traslade unha copia da mesma ao local da federación.

Alí os empregados da federación rematan o proceso revisando as actas e introducindo os datos nunha ferramenta de xestión como as que se comentan no seguinte apartado, co fin de actualizar os resultados e a clasificación para finalmente publicalos na páxina web da federación.

2.2. Competidores no mercado

Como se comentaba anteriormente, existen diversas ferramentas para xestionar competicións, unhas más avanzadas ca outras pero igualmente funcionais e que se van a introducir a continuación, resaltando certas vantaxes e carencias das mesmas.

2.2.1. Follas de cálculo

A folla de cálculo é o sistema utilizado por excelencia para xestionar competicións.

Un sistema rudimentario pero funcional, algunas federacións combinan en certa medida follas de cálculo e bases de datos sinxelas para crear as táboas das clasificacións e dos resultados, utilizando funcións que permiten automatizar algunas tarefas como por exemplo o cálculo da clasificación dos equipos en función dos seus resultados ao longo da competición.

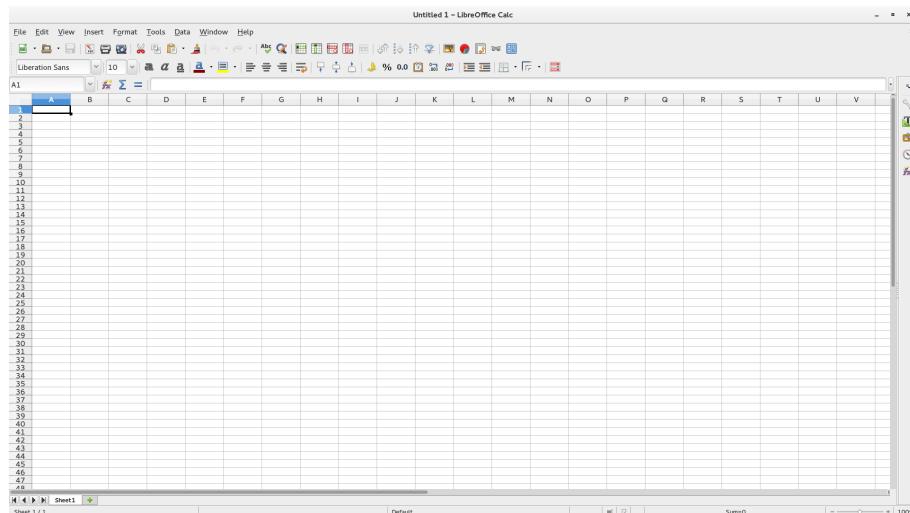


Figura 2.2: Folla de cálculo

O sistema é moi versatil para usuarios experimentados xa que permite adaptar de xeito sinxelo as posibles necesidades variables da federación, por exemplo na forma de calcular datos estadísticos, na forma en que se organizan as diferentes competicións, etc pero implica un gran traballo manual e pode chegar a ser un suplicio para usuarios con poucos coñecementos de ofimática.

As actas seguen a chegar en papel á federación e os datos deben ser introducidos nas diversas follas de cálculo (que en competicións de gran tamaño, vólvense inmanexables), revisando as sancións de xeito manual, polo que os errores na interpretación dos datos son habituais e por suposto os resultados non son mostrados en tempo real.

2.2.2. Ferramentas tradicionais

Tradicionalmente existiron varias ferramentas que comparten características de aplicacíons web clásicas xa que son solucións especializadas que deben ser instaladas individualmente para cada federación e personalizadas para cada unha a través dunha labor de consultoría.

A continuación imos presentar as dúas alternativas más extendidas no panorama nacional.

2.2.2.1. Novanet

Novanet é unha empresa especializada na xestión de competicións de fútbol e o seu producuto compónse únicamente dunha aplicación web desde a que crear as clasificacións pero tamén modificar as actas dos partidos ou ver os resultados.

No caso que nos incumbe, non dispón dunha aplicación que poida ser instalable nun teléfono móvil e os árbitros vense na obriga de acceder directamente a páxina web da federación para modificar as actas, pero que cando menos adáptase correctamente a terminais de menor tamaño como se pode observar na Figura 2.3.

Ademais, non permite o funcionamento do sistema de forma offline xa que non foi pensado inicialmente para o caso, o que obriga a levar a acta en papel para que se cubra igualmente e de esa forma, o árbitro poida subir posteriormente os datos a través da web cando chega a súa casa.

A interfaz é bastante complexa, o cal é un problema xa que a meirande parte dos árbitros son de avanzada idade e polo tanto resultalles difícil adaptarse aínda que está a mellorar.

Por último mencionar que únicamente está pensada para ser utilizada en fútbol e fútbol sala.

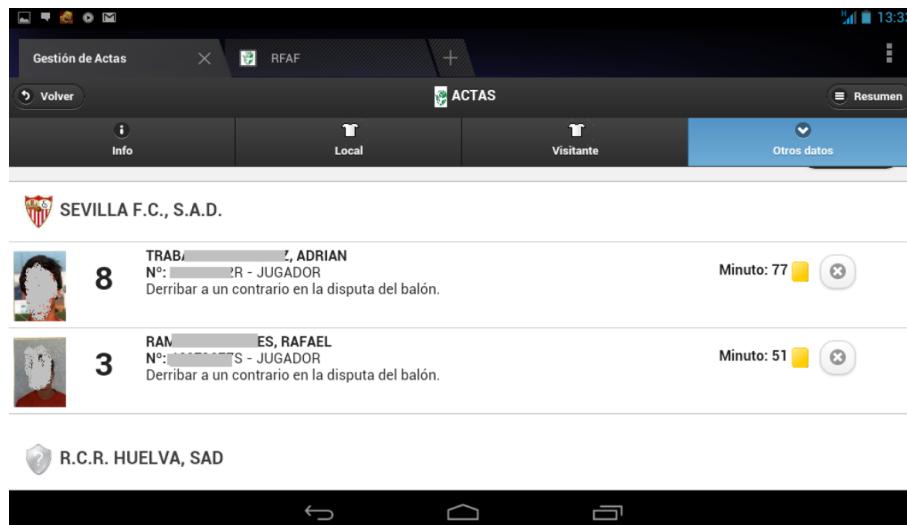


Figura 2.3: Aplicación web de Novanet

2.2.2.2. Federatio

O caso de Federatio é moi similar ao anterior, xa que tampouco dispón dunha aplicación específica para a xestión de actas electrónicas de forma sinxela, os árbitros deben acceder a través da páxina web ao chegar a casa para pasar os datos da acta física á versión electrónica.

A interfaz dista de ser atractiva xa que apenas se renovou dende que comezou a funcionar, entorno ao ano 2005, e non se atopa adaptada para terminais móbiles, o que dificulta enormemente a labor dos árbitros.

A pesar disto, é unha solución amplamente utilizada en deportes como o voleibol e o balonmano como se pode observar na Figura 2.4 na que podemos observar a páxina web da Federación Galega de Voleibol (FGVB) que dispón do sistema Federatio integrado.

LOCAL	VISITANTE	SET5
IES Politécnico	XAV Cambados B	25-22/23-25/16-25/26-24/16-14
CVP Gómez Lor Multípticas	Voleibol Mariñas	25-8/28-26/25-18

EQUIPO	PJ	PG	PP	NP	JE	JC	TF	TC	PUNTOS
1. CVP Gómez Lor Multípticas	4	4	0	0	12	0	303	180	8
2. IES Politécnico	5	2	3	0	10	12	443	480	7
3. XAV Cambados B	4	2	2	0	8	9	353	372	6
4. Voleibol Mariñas	5	1	4	0	5	14	367	434	6

Figura 2.4: Web da FGVB co sistema Federatio

2.2.3. Ferramentas na nube

Nos últimos anos xurdiron unha serie de tecnoloxías que democratizaron o acceso á computación na nube e por iso apareceron varios produtos específicos e centrados en xestionar dende unha única plataforma, múltiples federacións e asociacións deportivas.

2.2.3.1. miLeyenda

miLeyenda é unha de estas plataformas para a xestión de competicións na nube que permite aos administradores de federacións dispor tamén dunha aplicación móvil nativa para IOS e outra para Android que se pode ver na Figura 2.5.

Dende esta aplicación poden xestionar gran parte dos parámetros das súas competicións entre os que se atopan as clasificacións, as altas de xogadores e por suposto, as actas dos encontros.

Así mesmo tamén dispoñen dunha aplicación para que os xogadores e clubes poidan ver os resultados e as clasificacións polo que o custe de mantemento ditas aplicacións multiplicase por catro ao ter que soportar ata tal número de apps móbiles diferentes.

Esta é unha das grandes vantaxes de utilizar as tecnoloxías web que emprega VACmatch Mobile, permitindo utilizar unha soa aplicación para calquera sistema operativo e que incluso pode funcionar en un simple navegador web.

A usabilidade das aplicacións é tamén salientable e permite a xestión de diversos deportes pero a cambio non permite de momento cubrir as actas de forma offline, un problema habitual ante a falta de cobertura nos diversos pavillónns e campos deportivos.



Figura 2.5: App móvil de miLeyenda

2.2.3.2. Esportics

Esportics é outra solución tamén española que se centra na xestión de competicións deportivas de tenis, padel e deportes electrónicos e que lles permite aos xestores crear o seu propio espacio dentro do portal como o exemplo que temos na Figura 2.6.

A pesar de que a aplicación funciona para múltiples deportes, a súa adaptación é bastante forzada en certos menús e á hora de estruturar as competicións.

Únicamente dispón dunha páxina web adaptable a móbiles, a adaptación é tamén mellorable e non dispón dunha aplicación específica para que os árbitros poidan cubrir as actas dende o seu teléfono.

A usabilidade é aceptable, pero si engade unha complexidade que non é precisa en certos menús e que provoca que os árbitros de avanzada idade, lles resulte tamén pouco intuitivo.

Jornada/Ronda	Horario	Pista	Jugadores	Categoría	Resultado
Jornada 1 28/09/15 20:30	-	Lluís Martínez Manils - David Pages	VS	Robert Delgado - RAUL BERENGUEL MARTINEZ	Masculino Segunda 4-6 6-7
Jornada 1 28/09/15 21:00	-	Alex Butjosa Novales - David Sánchez Vallecillos	VS	Artur Costa - Pedro Lazaro Miguel	Masculino Segunda 6-4 4-6 5-7
Jornada 1 29/09/15 09:30	-	Oscar Bermudez - Christian Palomares	VS	Adrià Marin Chaparro - Arnau Inserfe Poyatos	Masculino Open 6-3 1-6 6-4

Figura 2.6: Torneo de padel na web de Esportics

2.2.3.3. Sportngin

Sportngin é unha solución integral para a xestión deportiva, probablemente un dos proxectos de referencia xa que dispón de aplicacíons web e móvil para a xestión e a visualización de competicións.

De feito, Sportngin permite a personalización da aplicación móvil para cada federación, cos seus logotipos, colores corporativos e incluso de certos menús, dando a posibilidade de cubrir as actas de forma offline, todo con unha interfaz moi amigable como a que podemos ver na Figura 2.7.

A parte das funcións habituais para xestionar as competicións e a creación de actas, cunha usabilidade moi coidada, tamén aporta como valor engadido a visualización e xestión de noticias,

fotos ou estadísticas da competición.

Por último, permite aos entrenadores planificar adestramentos e incluso comunicarse cos seus xogadores a través da mensaxería interna.

É sen dúbida o proxecto máis completo e a referencia a seguir a pesar de non ser software libre.



Figura 2.7: App móvil de Sportngin

2.2.4. Táboa comparativa

Para comparar dun xeito máis concreto as diversas alternativas creouse o Cadro 2.1 no que se poden observar as diversas características das aplicacións analizadas.

	Folla cálculo	Novanet	Federatio	miLeyenda	eSportics	Sportngin
Flexibilidade de configuración	Verde	Roja	Roja	Amarilla	Roja	Verde
Web responsive	Roja	Verde	Roja	Verde	Roja	Verde
Resultados en tempo real	Roja	Roja	Amarilla	Verde	Amarilla	Verde
Vista autoconfigurable	Roja	Roja	Roja	Verde	Roja	Verde
Aplicación móvil	Roja	Roja	Roja	Verde	Roja	Verde
Actas offline	Roja	Roja	Roja	Roja	Roja	Verde

Cadro 2.1: Táboa comparativa das alternativas existentes.

Os casilleiros en vermello indican que a solución non dispón da correspondente funcionalidade, en verde en caso de implementala e en amarelo para casos nos que non se dispón de información ou a solución utilizada é parcial.

2.2.5. Outras plataformas

Existen moitas ferramentas para a xestión de competicións e resultados pero apenas ningunha facilita aos árbitros unha plataforma sinxela e cunha usabilidade coidada.

Tamén existen pequenos plugins coa idea de extender outras plataformas xenéricas para adaptalas a xestión de competicións como o *Joomla! CMS sport extension*² pero a función final é moi limitada.

Por último temos outras propostas como *Siguetuliga*³ que permite que as persoas que se atopan vendo o partido, poidan subir os resultados pero simplemente é un complemento, non facilita nin elimina o traballo dos xestores de competicións.

2.3. Aplicacións libres no mercado

O software libre é un sector en crecemento na actualidade, os proxectos colaborativos que forman o mundo *Open Source* estanse a impor en múltiples mercados fronte ás correspondentes alternativas privativas que adoitan a ser tremendamente custosas e que atan ao usuario a tecnoloxías pechadas, impedíndolle ser o dono real do seu software, sen poder adaptalo ou engadirlle novas funcionalidades, nin tan sequera ver cómo se atopa feito e poder certificar así a seguridade da súa información.

Mesmo as grandes compañías TIC están a apostar por liberar parcial ou totalmente as súas tecnoloxías e produtos, favorecendo un desenvolvemento colaborativo fronte a idea atrasada do individualismo do modelo produtivo tradicional.

²Plugin para o sistema de xestión de contidos Joomla.

³Rede social para o deporte aficionado.

Concretamente no mundo do deporte é tremendamente complicado atopar algúns exemplo de aplicación baseada en software libre e as poucas existentes como *zuluru* ou *phpmysport* non destacan pola súa usabilidade nin polas súas funcionalidades, moi por detrás doutras solucións comerciais como as comentadas anteriormente e por suposto sen ningún tipo de aplicación móvil para facilitar a xestión das actas polo que é importante propor unha alternativa como VACmatch Mobile ás aplicacións privativas.

2.4. Solución aberta e adaptable

Os xestores de competicións habitualmente realizan unha considerable inversión económica para que unha empresa de consultoría lles cree unha aplicación web ou de escritorio a medida para a súa xestión. Algunhas mesmo dispoñen de aplicacións móbiles para os árbitros pero que son específicas para dito sistema de xestión polo que a reutilización de aplicacións non é posible.

Ademáis, estos sistemas son propietarios e o código non se atopa accesible polo que é imposible tratar de adaptar ditas aplicacións para outros sistemas de xestión ou engadirlle funcionalidades sen depender da empresa que comercializa o software.

É por isto polo que se chegou a conclusión de que é preciso crear unha plataforma aberta como é VACmatch Mobile que porporciona unha aplicación software libre adaptable a diversos deportes e que integra unha API⁴ de comunicacións aberta para a xestión de actas electrónicas e que permite a súa integración noutros sistemas de xestión entre os cales se atopa o sistema de VACmatch Web, unha implementación libre para a xestión de competicións.

⁴Application Programming Interface.

Capítulo 3

Metodoloxía

Índice xeral

3.1. Lean Startup	17
3.2. eXtreme Programming	17
3.3. Scrum	17
3.4. Adaptación da metodoloxía	18
3.4.1. Desenvolvemento orientado ao cliente	18
3.4.2. Sprints con backlog adaptable	18
3.4.3. Reunións semanais	18
3.4.4. Reunións diarias	19
3.4.5. Releases	19
3.4.6. Simplicidade	19
3.4.7. Tests	19
3.4.8. Fluxo de contribución ao proxecto	20

NESTE capítulo imos analizar as diversas metodoloxías utilizadas para a xestión do proxecto e explicar a adaptación das mesmas que finalmente se utilizou.

Comezaremos falando da metodoloxía orientada ao modelo de negocio xa que, como se indica na introdución, este proxecto xurdíu dentro de unha iniciativa empresarial e polo tanto dende o primeiro momento traballouse orientado cara o cliente, facendo que sexa nun dos piares do desenvolvemento.

Seguidamente comentaremos as metodoloxías áxiles que serviron de base para definir a metodoloxía utilizada finalmente, unha adaptación das mencionadas e que tamén se indica para rematar o capítulo.

3.1. Lean Startup

Lean Startup [Rie11] é unha metodoloxía para abordar o lanzamento de negocios e produtos a través da validación, a experimentación e a iteración no lanzamento dos mesmos co fin de acortar o ciclo de desenvolvemento.

É unha metodoloxía de traballo moi habitual nas startups que se centra na idea de *Crear - Medir - Aprender*, desenvolvendo pequenos produtos e realizando tests de mercado reais con verdadeiros clientes co fin de medir o seu grao de satisfacción e aprender para mellorar o producto en seguintes iteracións.

Habitualmente céñtrase na idea de crear un MVP (Minimum Viable Product), unha versión do producto que permite os desenvolvedores recoller co mínimo esforzo a máxima cantidade de coñecemento validado por parte dos clientes, evaluando as hipóteses de se os clientes realmente estarían dispostos a pagar polo producto e implicando a dito cliente no desenvolvemento do producto.

3.2. eXtreme Programming

eXtreme Programming [SG15] é unha metodoloxía de desenvolvemento áxil e incremental baseada na integración do cliente no desenvolvemento así como na simplicidade do código.

A metodoloxía apostá por facer as cousas sinxelas, sen preocuparse por ter que facer un pequeno traballo por adaptalas se é preciso, fronte a idea tradicional de facer un gran traballo para quizás nunca chegar a utilizar parte do mesmo.

As entregas funcionais son frecuentes e outras características como a importancia de introducir a programación en parellas para reducir o número de errores que se producen ao programar.

Por último aboga por introducir o TDD (Test Driven Development) [Mar08], implementando primeiro os tests, verificando que fallan para a continuación implementar o código que fai que pasen correctamente os mesmos. A idea é que os requisitos sexan convertidos a probas e deste modo cando os tests se pasen, poderemos garantizar que o código cumple os requisitos.

3.3. Scrum

Scrum [DGG12] tamén é unha metodoloxía incremental de desenvolvemento cunha serie de roles definidos para o proceso, cada un coas súas responsabilidades e que divide o proxecto en varios *Sprints* que son ciclos de desenvolvemento.

Cada un deles ten unha duración definida polo equipo de, habitualmente, entre unha e catro semanas, proporcionando un incremento de software entregable ao final de cada *Sprint*.

A totalidade das tarefas do proxecto atópanse definidas e priorizadas nunha lista chamada *Product Backlog*. Para cada sprint, selecciónanse aquellas tarefas que determinarán a lista a implementar durante a presente iteración, o *Sprint Backlog*, e que non pode variar ata rematar o sprint.

Durante todo o ciclo de traballo realizanse reunións diarias para comprobar o estado do proxecto así como outras ao finalizar e ao comezar os sprints, co fin de analizar a iteración anterior e planificar a seguinte, facendo un seguimento continuo do proxecto e facilitando a adaptación do mesmo a posibles novos requisitos.

3.4. Adaptación da metodoloxía

3.4.1. Desenvolvemento orientado ao cliente

O proxecto ten lugar dentro dunha iniciativa empresarial polo que se decidiu utilizar un modelo de desenvolvemento orientado ao cliente en todo momento, baseandose no pilar central da metodoloxía *Lean Startup*.

Para isto realizáronse diversas visitas as federacións para comprobar as súas necesidades a través dunha serie de entrevistas estructuradas para coñecer os problemas e a súa prioridade á hora de resolvelos.

Do mesmo modo realizáronse dous prototipos, un primeiro únicamente con plantillas HTML¹ para testear a organización da interfaz de usuario e un segundo xa funcional para comprobar a resposta dos usuarios finais ante o seu funcionamento.

3.4.2. Sprints con backlog adaptable

A organización do desenvolvemento organizouse de xeito moi similar á idea proposta en *Scrum*, dividindo o proceso en sprints, pequenas iteracións de díías ou tres semanas de duración e que cada unha proporciona unha serie de novas funcionalidades.

Cada sprint comeza cunha reunión de aproximadamente 30/45 minutos de duración na que realizar a planificación do mesmo en función do traballo realizado no sprint anterior, o que permite realizar melloras nas previsións segundo o aprendido dos anteriores.

A diferencia do proposto por Scrum, decidíuse optar por sprints de duración variable e cun backlog adaptable ao longo do sprint segúin as necesidades xa que proporciona unha maior flexibilidade e liberdade.

3.4.3. Reunións semanais

Todas as semanas faise unha reunión de 30/45 minutos de duración na que analizar o realizado na semana anterior e comprobar o seguimento da iteración co fin de atopar desviacións e corrixilas.

Cando unha reunión semanal coincide co fin de un sprint, dita reunión serve para realizar a planificación do seguinte sprint de xeito moi similar ás reunións de sprint que se realizan en *Scrum*.

¹HyperText Markup Language.

3.4.4. Reunións diarias

Ao comezar o día realiza unha análise duns 10 minutos de duración para revisalo realizado no día anterior e planificar de forma máis concreta o que se vai facer ese mesmo día.

3.4.5. Releases

Durante o desenvolvemento do proxecto trátase de aplicar a idea de realizar unha serie de pequenos entregables en cada iteración.

Todas as entregas ao finalizar unha iteración son totalmente funcionais pero non todas son versións entregables reais para ser postas en produción.

Durante o desenvolvemento producíronse 4 entregas (*releases*) totalmente funcionais, a primeira foi un prototipo, a segunda foi a versión real do proxecto, a terceira incorporou os tests e a cuarta correxiu diversas características para asegurar unha primeira versión estable.

3.4.6. Simplicidade

Utilizouse o principio de simplicidade que promove *eXtreme Programming* durante todo o desenvolvemento baixo a máxima de implementar únicamente o imprescindible en cada momento, sempre pensando en programar para hoxe e non para mañá.

A idea fundaméntase en realizar refactorizacíons de código para engadir novas funcionalidades a medida que son necesarias en lugar de invertir demasiado tempo na planificación e implementación de funcións que se supoñen necesarias e, algunas das cales, é probable que non sexan utilizadas finalmente.

3.4.7. Tests

A importancia de creación de tests automatizados está totalmente demostrada, atopándose en auxe metodoloxías como TDD (Test Driven Development)² ou BDD (Behaviour Driven Development)³ que tratan de dirixir o desenvolvemento a través dos tests e que son realizados antes da implementación da funcionalidade.

Durante a primeira parte do desenvolvemento non se aplicou ningunha destas metodoloxías pero a partir da terceira *release* e da integración dos primeiros tests, decidíuse optar por aplicar TDD no desenvolvemento, realizando probas unitarias nos servicios utilizados.

²Desenvolvemento dirixido polos tests.

³Desenvolvemento dirixido polo comportamento

3.4.8. Fluxo de contribución ao proxecto

O fluxo de traballo utilizado dende o primeiro día trata de simular o traballo diario de equipo e permite controlar a evolución do código de xeito máis ordenado.

Dispónse de unha rama *master* na que se atopa a versión estable de desenvolvemento así como de unha rama *development* que é máis inestable e que ao final de cada sprint, é integrada dentro de *master*.

Unha nova funcionalidade ou erro é resolto nunha nova rama independente, creada a partir de *development* e tratando que todas estas novas funcionalidades sexan independentes entre si.

Así mesmo tratase de que todos os *commits* sexan funcionais e o máis independentes posibles, evitando ter algúns que non compile ou que non pase os tests.

Posteriormente realizase unha *pull request*⁴ a través do mecanismo que proporciona o repositorio de código de GitHub, esperando que alguén revise o código para ser integrado na rama de desenvolvemento.

Cada certo tempo revisánse as *pull requests* abertas, analízase o código e se todo é correcto, acéptase.

Ao final de cada sprint realiza unha nova *pull request* para integrar a funcionalidade creada no sprint actual e que se atopa na rama de desenvolvemento, dentro da rama estable.

Dende a introducción de tests no proxecto, todo código subido ao repositorio é analizado a través dun sistema de integración continua —que se comenta con máis detalle na Sección 7.9.3,— e que comproba se os cambios engadidos pasan os tests ou non, e avisan por correo electrónico do resultado.

⁴Unha *pull request* é unha petición para integrar unha rama de Git en outra a través de GitHub, un mecanismo habitual para engadir novas funcionalidades ou correxir errores.

Capítulo 4

Análise de requisitos globais

Índice xeral

4.1. Consultas a xestores de federacións	21
4.2. Peticións obtidas	22
4.3. Requisitos finais	23
4.3.1. Usuarios	23
4.3.2. Listar actas	23
4.3.3. Visualizar actas	23
4.3.4. Xeración de actas offline	23
4.3.5. Modificación de actas	24

NESTE capítulo exporemos o proceso de análise de requisitos para o desenvolvemento do proxecto, explicando as diversas visitas que se realizaron a múltiples federacións.

Durante meses traballouse da man de varias destas federacións e asociacións deportivas tratando de comprender, non só as necesidades reais dos clientes se non tamén traballando na usabilidade da aplicación da súa man.

Comezaremos vendo as suxerencias recibidas das diversas federacións e finalmente expoñeremos os requisitos que finalmente se decidiron engadir ao proxecto.

4.1. Consultas a xestores de federacións

Para a realización deste apartado decidiuse consultar con diversas asociacións deportivas e federacións das que obter suxerencias e peticións acerca das necesidades que actualmente están a demandar, co fin de obter certas funcionalidades a implementar e incluso a priorización segundo as súas necesidades máis urxentes.

Na realización deste apartado contouse coa colaboración das asociacións e federacións deportivas que se detallan a continuación.

Tras varias reunións con eles, obtívose unha lista de requerimentos e suxerencias respecto das súas necesidades, que están detalladas na Sección 4.2, e dos que finalmente foron destilados os requisitos finais, que son presentados como Sección 4.3 de este capítulo.

Asociación de Peñas de Fútbol de A Coruña. É a asociación máis interesada polo proxecto e coa que se leva colaborando dende o primeiro momento, aportando suxerencias e incluso novos colaboradores para poder desenvolver un producto de calidade.

Realizáronse ata 5 visitas á federación co fin de mostrarllles a evolución do proxecto, comprobar a usabilidade da aplicación e o estado das funcionalidades.

UPOFU. A Asociación de Peñas ten boa relación coa UPOFU polo que nos facilitou o seu contacto e ofrecéronse da mesma maneira a colaborar co proxecto, interesados tamén en incorporalo na súa xestión.

Torneo VACmatch. Organizouse un torneo de fútbol sala co fin de testear o primeiro prototipo do proxecto con usuarios reais, tanto árbitros como xogadores e no que se comprobou as dificultades dos usuarios e se verificou a súa necesidade de dispor deste tipo de ferramentas.

Outras. Tamén se realizaron visitas a outras federacións incluso de outros deportes como o voleibol para comprobar os seus problemas na xestión e verificar a importancia de que o proxecto sexa facilmente adaptable a outros deportes.

4.2. Peticións obtidas

Cubrir acta en tempo real. A aplicación móvil debe permitir cubrir as actas e actualizar os resultados en tempo real co fin de manter a web da federación actualizada en todo momento.

Permisos. Débese dispor dun sistema de permisos para diferenciar a árbitros e outros xestores da competición.

Sincronización. A aplicación móvil debe sincronizar os datos coa plataforma central onde se atopa o sistema de xestión da federación e a súa web.

Persoas convocadas. É preciso poder dispor de tódalas persoas inscritas nun equipo e poder indicar de xeito sinxelo si esas persoas están ou non no encontro.

Eventos. A aplicación debe poder crear novos eventos, borrarlos e mostralos de xeito sinxelo e ao mesmo tempo de xeito xenérico, permitindo integrar calquera deporte.

Motivación dun evento. Debe poderse incluir en certos eventos un motivo polo que se creou, dispoñendo dunha lista de motivos por defecto e incluso permitindo ao xestor da federación, engadir novos motivos personalizados para a súa federación.

Editar dorsal dun xogador. Xa que en moitas competicións un xogador pode xogar cada partido con un dorsal diferente, debe poder cambiarse o dorsal por defecto dende a aplicación móvil.

Persoa con varios roles. Debe terse en conta a posibilidade de que unha persoa poida ter varios roles, tanto de xogador como de entrenador dentro de un equipo.

4.3. Requisitos finais

4.3.1. Usuarios

- **Facer login e logout.** A aplicación móvil debe permitir iniciar e pechar sesión para os árbitros.
- **Permisos para edición de actas.** A aplicación de xestión disporá de permisos diferenciados para editar as actas xa que os árbitros únicamente poden editar as actas que teñen asignadas.

4.3.2. Listar actas

- **Visualizar próximas actas a cubrir dun árbitro.** Mostrar a lista de próximas actas que ten para cubrir un árbitro, mostrando o lugar e a data do mesmo para facilitar o seu traballo.
- **Visualizar actas cubertas dun árbitro.** Debe mostrar as actas cubertas anteriormente e tódolos seus datos.
- **Actualización automática de actas descargadas ante modificacíons.** As actas deben actualizarse de forma automática na aplicación do árbitro unha vez o xestor da federación realiza a asignación dun partido a un colexiado.

4.3.3. Visualizar actas

- **Listar o personal e xogadores dun equipo.** Móstrase o personal e os xogadores do equipo na aplicación móvil.
- **Visualizar datos xerais dun acta.** Débese mostrar do xeito máis simplificado posible os datos xerais da acta nunha pantalla inicial para facilitar que sexa cuberta interactivamente durante o desenvolvemento do encontro.
- **Visualizar eventos dun acta.** Permitirse visualizar os eventos ordenados cronolóxicamente para facilitar a súa consulta.

4.3.4. Xeración de actas offline

A aplicación debe permitir a creación de actas de forma offline xa que pódese dar o caso de que a aplicación móvil non actualice as novas actas e o árbitro se vexa na obriga de crear unha acta de forma manual.

4.3.5. Modificación de actas

- **Modificación de propiedades da acta.** Débese permitir modificar propiedades da acta tales como a localización do encontro ou a data do mesmo.
- **Convocar un xogador ou entrenador.** A aplicación móvil permitirá indicar qué personal dos equipos están presentes no encontro así como editar certos datos dos mesmos como o seu dorsal.
- **Engadir un xogador que non está no equipo.** Pode darse o caso de que a un xogador débeselle permitir xogar un encontro aínda que non fose dado de alta na federación correspondente polo que é preciso poder engadir novos xogadores.
- **Editar datos de persoal creado.** Débese permitir editar certos datos dun xogador que foi creado dende a aplicación móvil como o nome, o dorsal ou o equipo o que pertencen.
- **Poder engadir motivos dun evento xerado.** A federación debe poder engadir novos motivos personalizados para poder engadir a un evento dende a aplicación de xestión.
- **Cambiar de parte.** A aplicación móvil debe permitir cambiar de parte no encontro.
- **Modificar o tempo.** A aplicación móvil disporá dun cronómetro que permita seguir o tempo do encontro así como permitirá modificalo manualmente por si hai algún desaxuste durante o encontro.
- **Engadir observacións na acta.** O árbitro debe poder engadir observacións as actas dos encontros.
- **Asinar a acta.** Tanto persoal do equipo como árbitros deben poder asinar as actas con un código PIN do que disporá cada un.
- **Engadir eventos deportivos.** A aplicación debe facilitar a adaptación de novos deportes e a posibilidade de engadir de xeito sinxelo novos eventos.

Capítulo 5

Planificación e seguimento

Índice xeral

5.1. Validación de negocio (Xullo 2015 – Novembro 2015)	26
5.1.1. Prototipo visual	26
5.1.2. MVP funcional	27
5.2. Desenvolvemento de produto (Novembro 2015 – Xaneiro 2016)	29
5.2.1. 1 ^a e 2 ^o iteración. Creación do proxecto e xestión de actas	30
5.2.2. 3 ^a iteración. Eventos	32
5.2.3. 4 ^a iteración. Xestión de usuarios e creación offline de actas	34
5.2.4. 5 ^a iteración. Sinaturas	36
5.3. De empresa a comunidade (Xaneiro 2016 – Maio 2016)	36
5.3.1. 6 ^a e 7 ^a iteración. Optimización e melloras	37
5.3.2. 8 ^a iteración. Testing e integración continua	38
5.3.3. 9 ^a e 10 ^a iteración. Inxección de dependencias	39
5.3.4. Release 0.2.0: Usabilidade en menús	41
5.3.5. Release 0.2.1: I18n e app híbrida	43
5.3.6. Release 0.2.2: Finalización da memoria e revisión de errores	44

NESTE capítulo detallaremos a planificación e o seguimento do proxecto, un proxecto que por diversas circunstancias se dividiu principalmente en tres grandes etapas, as dúas primeiras mentres VACmatch era unha iniciativa emprendedora baseada nun proxecto software libre e a última durante a cal se comezou unha conversión da iniciativa cara un proxecto comunitario.

Durante o desenvolvemento xuridiron tamén diversos acontecementos relacionados co proxecto que tamén é importante resaltar debido a súa influencia no desenvolvemento e que se comentan durante este capítulo.

Agosto 2015 - Outubro 2015 VACmatch. Validación de negocio.

Outubro 2015 - Xaneiro 2016 VACmatch. Desenvolvemento de producto.

Xaneiro 2016 - Xuño 2016 De empresa a comunidade.

5.1. Validación de negocio (Xullo 2015 – Novembro 2015)

A duración desta etapa é de aproximadamente 4 meses e ven determinada polos primeiros pasos de VACmatch como iniciativa empresarial e que levan a orientar o desenvolvemento do produto cara o cliente, comezando cunha serie de prototipos para coñecer as suas necesidades e validar a idea de negocio.

Durante o primeiro mes planifícase a realización dun prototipo visual co fin de comprobar a usabilidade e consolidar os requisitos dos clientes. De seguido, plantéxase crear un pequeno prototipo funcional, un MVP¹ na metodoloxía Lean Startup, co obxectivo de testear as necesidades dos clientes e definir o producto final a desenvolver.

En total realizanse dúas versións do prototipo da aplicación que son analizadas a continuación.

5.1.1. Prototipo visual

5.1.1.1. Planificación e definición da iteración

Esta iteración dura un total de 4 semanas de desenvolvemento entre o 15 de Xullo e o 16 de Agosto e realizase unha visita semanal ao cliente para obter feedback e mostrarlle a evolución do prototipo.

Durante este periodo planificouse o desenvolvemento dunha aplicación moi sinxela e sen funcionalidade, que únicamente permite analizar a usabilidade do sistema e comprobar se é factible adaptar o proceso de creación de un acta deportiva nunha aplicación móvil.

Así mesmo, ao longo do período realizaranse ata tres visitas á federación coa que se traballou dende o primeiro momento para comprobar a experiencia de un futuro usuario real da aplicación e obter feedback para futuras melloras.

5.1.1.2. Revisión e feedback

Durante as visitas as federacións obtivéreronse diversas prospostas que levaron a adaptar o prototipo, algunas das cales se mencionan a continuación:

- Facer interactiva a aplicación e non mostrar grandes táboas con datos.
- Todas as accións deben xirar ao redor da acta.
- Crear partidos cando non hai cobertura.

5.1.1.3. Tarefas e seguimento

A descomposición das tarefas desta iteración son as seguintes:

¹ Mínimo Producto Viable

- V.1** Crear esqueleto da aplicación.
- V.2** Como árbitro quero poder consultar as próximas actas a cubrir.
- V.3** Como árbitro quero poder consultar as actas xa cubertas.
- V.4** Como árbitro quero poder editar un acta.
- V.5** Como árbitro quero poder ver un acta.
- V.6** Como árbitro quero poder ver os xogadores de ambos equipos.
- V.7** Como árbitro quero poder engadir un evento.
- V.8** Como árbitro quero poder rematar ou suspender un partido.
- V.9** Como árbitro quero poder logearme.
- V.10** Como árbitro quero poder seleccionar cales xogadores de cada equipo se atopan no encontro.
- V.11** Como árbitro quero poder borrar un evento.
- V.12** Estudio sobre React e Flux.

Para o desenvolvemento desta primeira versión do prototipo planificáronse 80 horas e puidéronse realizar tódalas tarefas no tempo indicado.

5.1.2. MVP funcional

5.1.2.1. Planificación e definición da iteración

Esta iteración dura un total de 3 meses e desenvólvese entre o 16 de Agosto e o 15 de Novembro, realizando múltiples visitas a federacións e asociacións deportivas.

Unha vez finalizadas as probas visuais e de usabilidade procédese a planificar o desenvolvemento para adaptar o prototipo e engadirlle funcionalidade sinxela, sen validacións e sen funcionalidade offline, co obxectivo de obter un prototipo funcional que poida ser utilizado por usuarios reais nun entorno controlado.

Engadirase funcionalidade para as vistas creadas na iteración anterior, comezando polo listado de actas pendentes e rematadas e diversos compoñentes xenéricos como os que se utilizan para listar xogadores e outros elementos como poden ser as actas.

Únicamente se engadirá a funcionalidade básica imprescindible para xestionar un encontro, excluindo requisitos como a sinatura de actas ou a creación offline das mesmas co fin de axilizar as primeiras probas.

Durante a iteración tamén se farán visitas á federación para mostrar o estado do desenvolvemento e para buscar que tamén árbitros reais vexan os progresos e proporcionen feedback.

Unha vez rematada a iteración realizarase o torneo onde probar o prototipo desenvolto nun caso real; pódese ver o desenvolvemento de dita competición na Sección 5.1.2.4.

5.1.2.2. Revisión e feedback

Durante as visitas as federacións obtivéronse múltiples propostas e melloras, moitas das cales será incorporadas ao backlog do proxecto mentres que outras serán rexeitadas polo momento ao non considerarse prioritarias ou por ser casos de usos moi concretos para esa federación e difficilmente extrapolables a outras.

Listaxe de tarefas engadidas ao backlog.

- Meter xogadores manualmente xa que poden non terse creado no sistema de xestión.
- Poder ver os eventos de forma sinxela dende a vista de fin de partido para que os equipos vexan o que están asinando.
- Editar dorsal dos xogadores xa que poden cambiar.
- Ter opción de non poñer motivo para as tarxetas.
- Mostrar foto de xogador ao engadir un evento.

Listaxe de tarefas rexeitadas polo momento.

- Ao marcar doble amarela, avisar da expulsión. Moi concreta para un deporte, non se implementa de momento.
- Mostrar confirmación de que se engadíu un evento.
- Avisar aos delegados/personal do clube cando se sube un acta.
- Posibilidade de que o árbitro engada un anexo na casa á acta en lugar de escribir as incidencias.

5.1.2.3. Tarefas e seguimento

As tarefas que se realizarán durante esta iteración son as seguintes:

MVP1 Como árbitro quero poder cargar a lista de actas pendentes.

MVP2 Como árbitro quero poder cargar a lista de actas rematadas.

MVP3 Como árbitro quero poder ver os datos dun partido.

MVP4 Como árbitro quero poder ver a lista de xogadores dun equipo.

MVP5 Como árbitro quero poder seleccionar os xogadores presentes no partido.

MVP6 Como árbitro quero poder engadir un gol a un xogador.

MVP7 Como árbitro quero poder engadir unha falta a un xogador.

MVP8 Como árbitro quero poder engadir unha tarxeta (amarela ou vermella) a un xogador.

MVP9 Como árbitro quero poder ver a lista de eventos de un partido.

MVP10 Como árbitro quero poder borrar eventos de un partido.

MVP11 Como árbitro quero poder engadir incidencias a un partido.

Planificouse un total de 65 horas para desenvolver esta iteración que non resultaron suficientes, obligando a facer un total de 10,5 horas extra motivado polo descoñecemento da tecnoloxía, de forma que foi preciso invertir un maior número de horas das esperadas en formación.

5.1.2.4. I Torneo VACmatch

Durante este periodo tamén se planificou a organización dun torneo de fútbol sala a finais do mes de Outubro coa idea de probar nun entorno real e controlado, os primeiros prototipos desenvoltos.

Finalmente participaron 6 equipos e mais de 50 persoas durante os dous días que durou o evento, obtendo todo tipo de suxerencias e detectando múltiples errores que serán analizados e comentados na seguinte iteración.

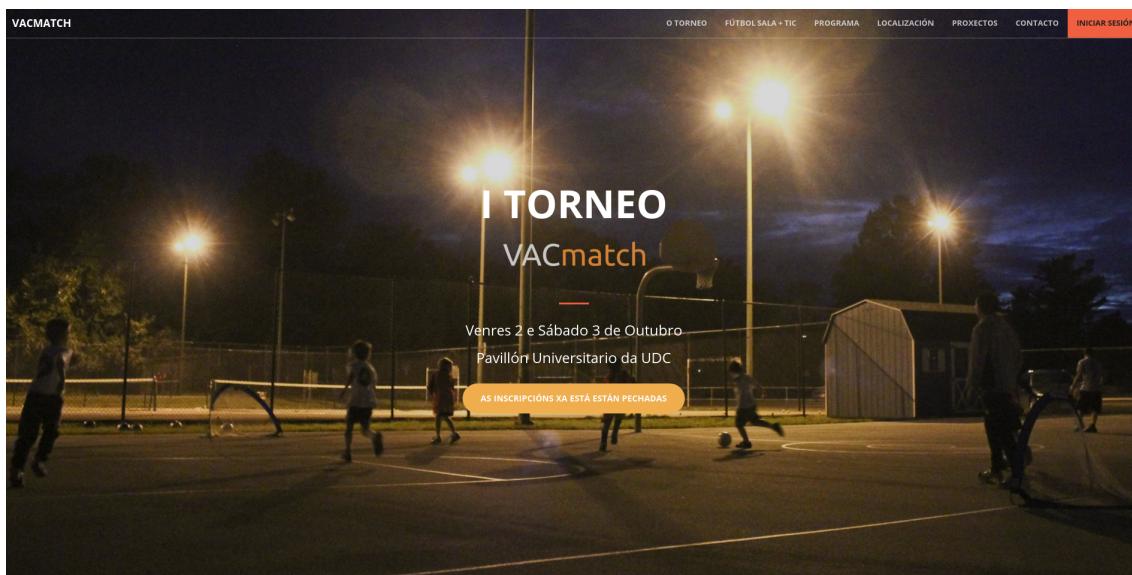


Figura 5.1: Web do I Torneo VACmatch

5.2. Desenvolvemento de produto (Novembro 2015 – Xaneiro 2016)

Tras analizar os problemas e os errores cometidos durante a realización do I Torneo VACmatch, e concretamente no funcionamento da aplicación móvil durante o mesmo, decidíuse comezar novamente dende o principio un proxecto novo en lugar de facer unha refactorización do prototipo.

Durante este tempo decidimos inscribir o proxecto no “Concurso Universitario de Software Libre” o que incentivou a comezar a escribir un blog técnico, a través da conta de Medium² de VACmatch, no que contar os avances que suceden durante o desenvolvemento do proxecto.

Neste periodo de 3 meses de duración planifícase adicar unha media de 4 horas diarias en VACmatch Mobile debido a que en paralelo se está a traballar na versión de VACmatch Web.

Así mesmo, descontando os múltiples días festivos, calcúlase un total de 208 horas de traballo divididas en 5 iteracións.

5.2.1. 1ª e 2º iteración. Creación do proxecto e xestión de actas

5.2.1.1. Planificación e definición da iteración

Estas iteracións transcorren ao longo do mes de Novembro, adicando 15 días a cada unha.

A primeira céntrase en comezar o desenvolvemento da aplicación pensando dende o primeiro momento no funcionamento tanto online como offline e controlando os posibles conflictos que poidan suceder entre os datos.

Comézase co estudo da tecnoloxía en profundidade xa que os prototipos non utilizaban nin Reflux nin PouchDB e simplemente enviaban os datos a un servizo web remoto.

crease tamén o proxecto base engadindo a licencia e defínese o modelo de datos, que vai cambiar bastante do modelo inicial, pensando para unha base de datos relacional que é a que podíamos atopar na API remota de VACmatch Web.

É por iso que os datos serán desnormalizados e pasarán a almacenarse en documentos en lugar de en táboas.

Por último comézase a implementación do listaxe de actas a cubrir e de un botón para engadilas e eliminarlas de xeito sinxelo para facer as primeiras probas.

Durante a 2º iteración planifícase o desenvolvemento das páxinas principais e básicas para a xestión da acta dun encontro.

As funcionalidades que se abordarán neste sprint céntranse principalmente na vista na que se mostra o resumo actual da acta así como o control do tempo, co fin de permitir xestionar o encontro en tempo real de forma interactiva.

5.2.1.2. Revisión e feedback

Durante esta iteración analizáronse os problemas detectados durante o torneo, optando por comezar a implementación do proxecto dende cero como se comentou anteriormente.

Tamén se realizaron diversas publicacións do blog para comentar a realización de dita competición e sobre todo analizar aspectos como a elección tecnolóxica, a metodoloxía de desenvolvemento

²Medium é unha rede social que permite crear e seguir blogs de múltiples temáticas

e as próximas funcionalidades a abordar.

Ao non realizar ningunha visita a federacións, non se obtivo o seu feedback.

5.2.1.3. Tarefas e seguimento

Durante estas iteracións realizáronse as seguintes tarefas que tamén se poden observar a través dos Diagramas de Gant [5.2](#) e [5.3](#).

S1.1 Definir modelo de datos.

S1.2 Definir arquitectura e tecnoloxía.

S1.3 Deseñar mockups.

S1.4 Crear proxecto base.

S1.5 Crear modelos en PouchDB.

S1.6 Estudo da tecnoloxía. React, Reflux, PouchDB, Redmine

S1.7 Como árbitro quero poder obter a lista de actas a cubrir.

S1.8 Permitir crear e borrar actas para tarefas de test.

S1.9 Engadir licencia e Readme

S2.1 Como árbitro quero ver o resumo da acta.

S2.2 Como árbitro quero controlar o tempo do partido.

S2.3 Como árbitro quero manter o tempo do partido aínda que cambie de páxina.



Figura 5.2: Diagrama de Gant do sprint 1

A planificación inicial de 71 horas de desenvolvemento finalmente cumpríuse correctamente, incluso reducindo o tempo empregado ata as 68.

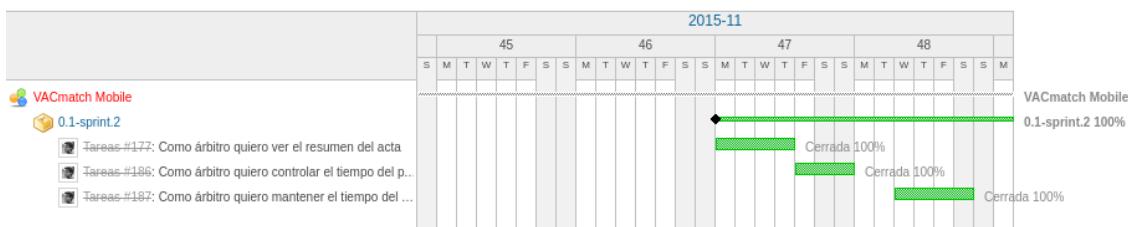


Figura 5.3: Diagrama de Gant do sprint 2

5.2.1.4. Participación na I Lonxa de Financiamento Responsable

Durante este tempo tamén cómpre destacar a participación de VACmatch na I Lonxa de Financiamento Responsable en Galicia, permitíndonos presentar o noso proxecto ante diversos inversores preocupados pola responsabilidade social das empresas.



Figura 5.4: I Lonxa de Financiamento Responsable

Foi unha experiencia única que nos permitiu introducirnos por primeira vez no mundo da inversión en startups e coñecer aos diversos proxectos que se presentaron, obtendo tamén feedback dos asistentes e participantes.

5.2.2. 3^a iteración. Eventos

5.2.2.1. Planificación e definición da iteración

Esta iteración comeza o 30 de Novembro e remata o 13 de Decembro, unha iteración de dúas semanas de duración que se centrada principalmente na xestión de eventos dos encontros, tarefas que requiren unha forte planificación e análise xa que se busca que ditos eventos sexan o máis

xenéricos posibles e facilmente adaptables aos diversos deportes.

Da mesma forma plantéxase crear tipos de eventos que modifiquen tamén a propia acta, actualizando na mesma tanto o resultado como as faltas cometidas.

Por último incorpórarse tamén na iteración a posibilidade de convocar e editar persoas dun equipo así como algúin pequeno erro detectado na última iteración.

5.2.2.2. Revisión e feedback

Esta iteración centróuse en avanzar a maior velocidade no desenvolvemento, pódese observar cómo o número de horas asignadas supera a media teórica planificada para o período completo.

Non se realizou ningunha visita a federacións e centrouse prácticamente todo o esforzo, fora do desenvolvemento, en preparar a presentación para a I Lonxa de Financiamento Responsable en Galicia da que se falou anteriormente na Sección 5.2.1.4 e na que se obtiveron diversos comentarios para enfocar o modelo de negocio e o desenvolvemento da aplicación cara o cliente.

5.2.2.3. Tarefas e seguimento

A continuación móstranse as diversas tarefas realizadas na iteración e pódese ver o Diagrama de Gant correspondente na Figura 5.5:

S3.1 Como árbitro quero poder engadir un evento.

S3.1.1 Como árbitro quero ver a lista de xogadores para asignar un evento.

S3.1.2 Como árbitro quero poder confirmar engadir un evento.

S3.1.3 Como árbitro quero engadir unha causa a un evento.

S3.2 Cómo árbitro quero poder ver a lista de eventos de un partido.

S3.3 Como árbitro quero que se xeneren eventos de comezo e fin de partido.

S3.4 Como árbitro quero que se xeneren eventos ao cambiar de parte.

S3.5 Como árbitro quero que se actualice o resultado ao engadir un gol.

S3.6 Como árbitro quero que se actualicen as faltas automáticamente.

S3.7 Como árbitro quero poder convocar e desconvocar xogadores.

S3.8 Como árbitro quero poder cambiar o dorsal de un xogador nun partido determinado.

S3.9 Como árbitro quero poder borrar un evento.

S3.10 Como árbitro quero ver a lista de eventos de un partido ordenada por tempo e parte.

S3.11 Mostrar únicamente xogadores convocados ao engadir un evento.

S3.12 Como árbitro quero que ao borrar un evento se actualicen os resultados e as faltas na Acta.

S3.12 Erro: Cando non hai ningún evento de cambio de parte hai un error.

S3.13 Erro: Correxir erro cando se engade un evento con causa.

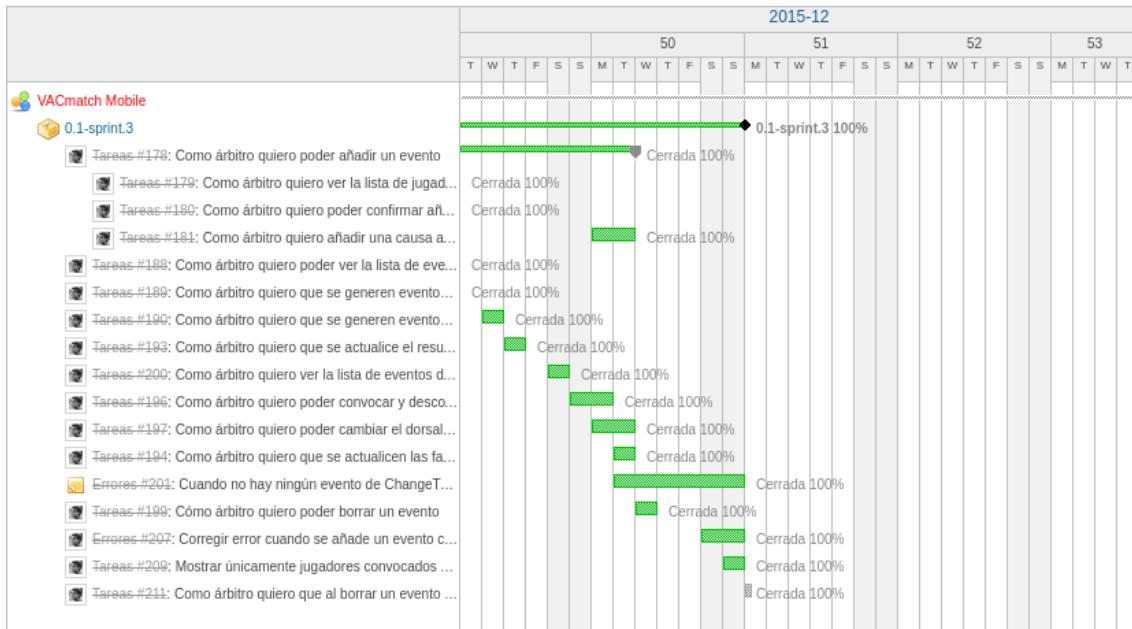


Figura 5.5: Diagrama de Gant do sprint 3

Finalmente a planificación inicial de 51 horas cumplíuse correctamente quedando un extra de 6 horas que foron reasignadas ao outro proxecto.

5.2.3. 4^a iteración. Xestión de usuarios e creación offline de actas

5.2.3.1. Planificación e definición da iteración

Esta iteración ten lugar entre o 14 e o 30 de Decembro de 2015 e céntrase na autenticación da aplicación que debe integrar unha base de datos remota para que os árbitros poidan conectarse co seu usuario e tamén inclúe a creación e edición manual de actas que en anteriores iteracións foi engadida únicamente para probas internas pero sen realizar as comprobacións necesarias.

Tamén se planifican tarefas para comenzar as primeiras partes da memoria do proxecto que incluen a introdución, o estado da arte e os fundamentos tecnolóxicos.

5.2.3.2. Revisión e feedback

Durante este periodo escribíuse unha nova entrada no blog do proxecto na que se fixo unha introdución aos fundamentos tecnolóxicos do mesmo, explicando a motivación da elección das tecnoloxías e mostrando un exemplo moi sinxelo.

Por último mostrouse tamén a estrutura da aplicación e as partes más importantes do proxecto, pensando en facilitar a introdución no desenvolvemento aos posibles interesados en colaborar no mesmo.

5.2.3.3. Tarefas e seguimento

O listado de tarefas abordadas durante esta iteración é o seguinte:

S4.1 Como usuario quero poder facer log in.

S4.2 Engadir diferenciación entre Persoal e Xogadores.

S4.3 Como usuario quero poder facer log out.

S4.4 Como árbitro quero poder crear un partido manualmente.

S4.5 [Memoria] Definir introdución.

S4.6 [Memoria] Estado da arte.

S4.7 [Memoria] Fundamentos tecnolóxicos.

S4.8 [Memoria] Engadir modelo.

S4.9 Como árbitro quero poder editar un acta creada dende o móvil.

S4.10 Erro: Correxir erro coa variable dialogIsOpen ao editar o dorsal dun xogador.

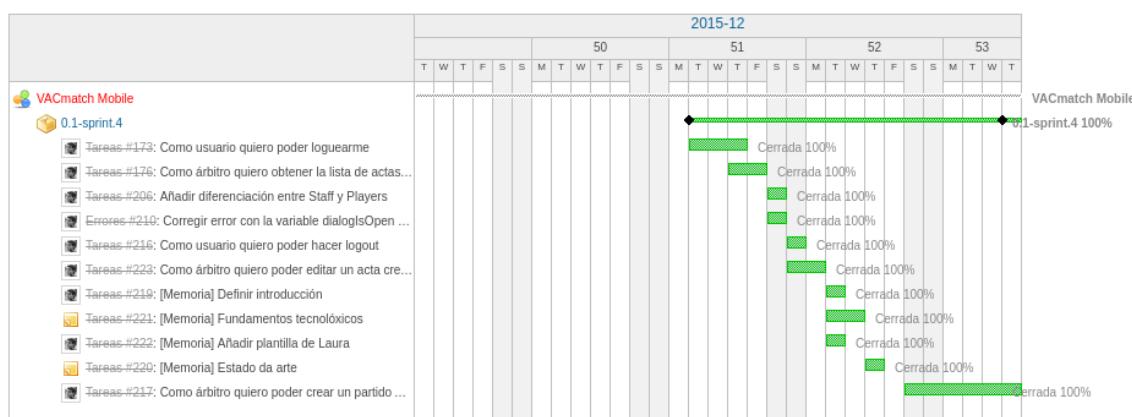


Figura 5.6: Diagrama de Gant do sprint 4

Esta vez a planificación non foi moi acertada debido a diversos imprevistos externos ao proxecto, o que levou a necesidade de mover diversas tarefas á seguinte iteración para axustar a planificación inicial de 53.5 horas ata as 44 que se realizaron finalmente.

5.2.4. 5^a iteración. Sinaturas

5.2.4.1. Planificación e definición da iteración

Esta iteración comeza o día 1 de Xaneiro de 2016 e remata o día 8 e centrase na última vista da aplicación, aquela que permite xestionar a finalización do encontro dando a posibilidade de asinar a acta e engadir comentarios por parte do árbitro.

Na iteración anterior detectáronse varias melloras a incluir como facer que por defecto as contas creadas dende a aplicación móvil sexan todas de tipo árbitro. Así mesmo facer que se engada a dito usuario como árbitro do encontro en tódalas actas que cree esa conta.

5.2.4.2. Revisión e feedback

Ao finalizar esta iteración publicouse unha nova entrada no blog do proxecto expoñendo a metodoloxía áxil utilizada para o desenvolvemento, así como aquelas nas que se basea (Scrum e eXtreme Programming) e a metodoloxía de negocio orientada cara o cliente (Lean Startup).

Por último tamén se publicou o fluxo de traballo que se realiza para contribuir ao proxecto co sistema de control de versións Git, co fin de facilitar o traballo a futuros contribuidores.

5.2.4.3. Tarefas e seguimento

Durante esta iteración realizáronse as seguintes tarefas e das cales se pode observar o seu diagrama de Gant na Figura 5.7.

S5.1 Como árbitro quero poder asinar un acta.

S5.2 Como árbitro quero que unha ou varias persoas convocadas de cada equipo poidan asinar un acta.

S5.3 Como árbitro quero poder engadir comentarios a un acta.

S5.4 Como árbitro quero poder borrar un xogador da lista de convocados.

S5.5 Crear un árbitro ao crear un novo usuario na aplicación móvil.

S5.6 Engadir o árbitro que ten o usuario asignado nas actas que crea.

Nesta iteración a planificación incial de 32 horas foi sobreestimada polo que finalmente sobraron 5 horas de desenvolvemento que foron reasignadas ao outro proxecto.

5.3. De empresa a comunidade (Xaneiro 2016 – Maio 2016)

Durante o mes de Xaneiro de 2016 decidíuse abandonar o proxecto de VACmatch como iniciativa empresarial por diversos motivos e continuar con él únicamente como proxecto comunitario de software libre.

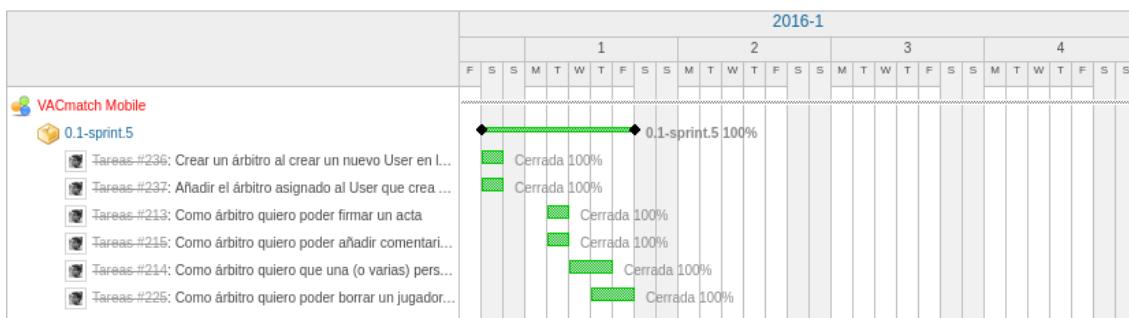


Figura 5.7: Diagrama de Gant do sprint 5

Motivado por isto, producíuse un parón de aproximadamente un mes de duración e ao mesmo tempo comezouse a traballar nunha empresa externa polo que durante este período diminuí considerablemente o tempo dispoñible para continuar co proxecto.

O número de horas total estimado para este período de 6 iteracións é de 189, todo nun total de 5 meses, o que supón unha media aproximada de 2 horas diarias de desenvolvemento.

5.3.1. 6^a e 7^a iteración. Optimización e melloras

Motivado pola inestabilidade xerada polos cambios mencionados anteriormente, non se realizou unha correcta planificación da 6^a iteración e finalmente non foi posible realizar ninguna tarefa polo que se decidiu unir ambas iteracións.

5.3.1.1. Planificación e definición da iteración

Estas iteracións comienzan o 14 de Xaneiro de 2016 e rematan o 29 de Febreiro, a pesar de que non se realizou traballo efectivo ata as últimas dúas semanas de Febreiro.

Durante este periodo planificouse unha importante refactorización de código co fin de simplificar certas partes do mesmo, facilitar o mantemento da aplicación e revisar a forma na que se crean os identificadores dos obxectos en base de datos.

5.3.1.2. Revisión e feedback

Como se comentou anteriormente, este foi un sprint marcado por un longo parón de aproximadamente un mes e medio durante o que o proxecto non avanzou nin se obtivo ningún tipo de feedback.

En cambio durante este tempo si se publicaron varias entradas no blog do proxecto co fin de poñer ao día de xeito público os últimos avances do mesmo e contar diversas decisións técnicas que se adoptaron como a selección da licencia para o proxecto, a implementación do sistema de eventos ou a sinatura das actas.

5.3.1.3. Tarefas e seguimento

Durante esta iteración realizáronse poucas tarefas, todas imputadas ao sprint número 7, pero dunha duración considerable como se pode observar no diagrama de Gant da Figura 5.8.

S7.1 Refactorizar servicios.

S7.2 Crear clases para cada entidade.

S7.3 Revisar cómo se crean os identificadores dos obxectos en base de datos.

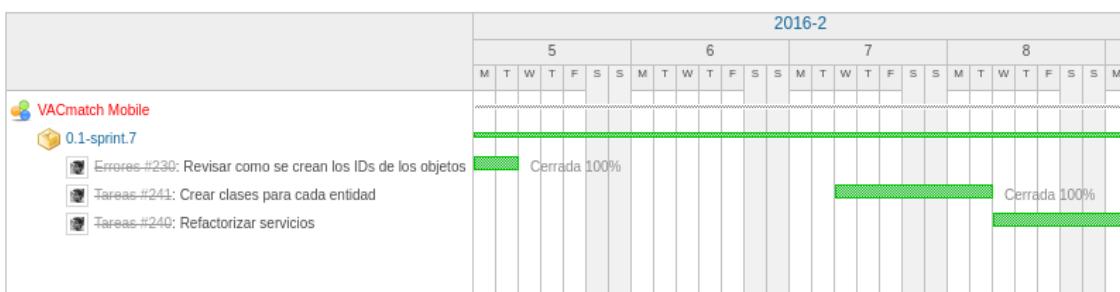


Figura 5.8: Diagrama de Gant do sprint 7

Para a realización desta iteración planificóuse un total de 22 horas e non se precisou ningunha hora extra para completar o traballo.

5.3.2. 8ª iteración. Testing e integración continua

5.3.2.1. Planificación e definición da iteración

A iteración transcorre entre os días 1 e 14 de Marzo.

Decidiuse engadir tests unitarios para previr futuros erros e facilitar o mantemento da aplicación xa que en todo proxecto de certa embergadura, e máis en proxectos libres nos que calquera pode colaborar, é importante asegurar que os novos cambios que se engadan non xeneren problemas no funcionamento da aplicación.

Relacionado con este tema tamén se planificou a integración do repositorio de código con unha ferramenta de integración continua que facilite a execución de este tipo de probas, en este caso Travis CI do que falaremos na Sección 7.9.3.

5.3.2.2. Revisión e feedback

Durante esta iteración foi preciso publicar unha folla de ruta do proxecto no blog do mesmo, requisito fundamental solicitado pola organización do Concurso Universitario de Software Libre no que VACmatch se atopa inscrito dende o mes de Outubro aproximadamente.

É por iso que se fixo unha entrada resumo para mostrar as tarefas existentes no Redmine do proxecto onde se realiza toda a xestión de incidencias así como se resaltaron os seguintes pasos a seguir no mesmo, como por exemplo, engadir a internacionalización ou permitir poñer en funcionamento a aplicación nun dispositivo móbil.

5.3.2.3. Tarefas e seguimento

As seguintes tarefas son as realizadas en esta iteración:

S8.1 Engadir tests aos servizos de Eventos, Persoas, Equipos e Actas.

S8.2 Engadir tests aos servizos de Auth, Árbitros e Sinaturas.

S8.3 Engadir integración continua con Travis CI.

S8.4 Engadir campos para confirmar contrasinal e código PIN ao crear un usuario.

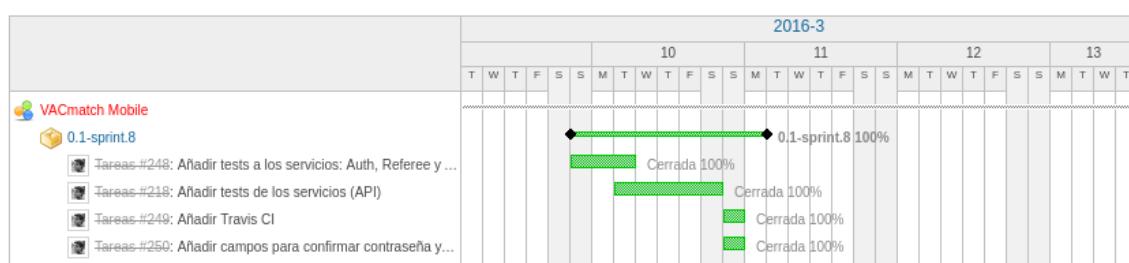


Figura 5.9: Diagrama de Gant do sprint 8

No diagrama da Figura 5.9 pódese observar a evolución das tarefas ao longo do sprint que inicialmente tivo unha planificación de 37 horas pero finalmente alongouse 16 horas extra, obligando a ampliar a xornada de traballo en 2 fines de semana a media xornada co obxectivo de non retrasar a execución das tarefas.

5.3.3. 9^a e 10^a iteración. Inxección de dependencias

Motivado de novo pola inestabilidade e a falta de tempo dispoñible para realizar o proxecto, finalmente decidíuse integrar de novo estas dúas iteracións en unha.

5.3.3.1. Planificación e definición da iteración

Estas iteracións comezan o día 15 de Marzo e rematan o 25 de Abril.

Detectáronse problemas en Travis xa que non detectaba errores nos tests polo que é o primeiro que había que corrixir.

Tamén se decidíu crear unha barra de notificacións compartida para tódolos compoñentes da aplicación e engadíronse estados diferentes para as actas co fin de mostrar cando un encontro non comezou, cando se está a xogar e cando rematou.

Pero a tarefa máis importante xurdíu ao aparecer un problema de dependencias circulares que obrigou a engadir unha factoría para realizar inxección de dependencias entre os servizos da aplicación xa que varios, dependían uns de outros.

Finalmente planificouse tamén a corrección de diversos erros detectados na iteración anterior como o feito de non poder eliminar un xogador dun partido sen ter eliminados anteriormente os eventos que ten asignados en ese partido.

5.3.3.2. Revisión e feedback

Durante esta iteración non se publicou ningunha entrada no blog e tampouco se recibiron suxerencias.

5.3.3.3. Tarefas e seguimento

As tarefas definidas para esta iteración son as seguintes:

S10.1 Engadir textos de error na aplicación.

S10.2 Crear barra de notificacións para utilizar en calquera compoñente.

S10.3 Erro: Correxir problema de dependencias circulares.

S10.3.1 Engadir DAOs.

S10.3.2 Engadir Inxección de Dependencias.

S10.4 Erro: Mostrar que o partido remató na acta.

S10.4.1 Engadir estados na Acta.

S10.5 Erro: Como árbitro non debería poder convocar/borrar a un xogador que ten eventos asignados.

S10.6 Erro: Correxir errores varios derivados de engadir inxección de dependencias.

S10.7 Erro: Un evento de comezo de partido non se pode borrar si existe algún outro evento creado.

S10.8 Erro na xestión do estado da Acta.

S10.9 Erro: Travis CI non funciona correctamente

A estimación inicial de horas para esta iteración foi de un total de 78, resultando unha vez más, optimista e permitindo que a diferencia de horas coas reais, 66, poidesen ser reasignadas ao proxecto de VACmatch Web.

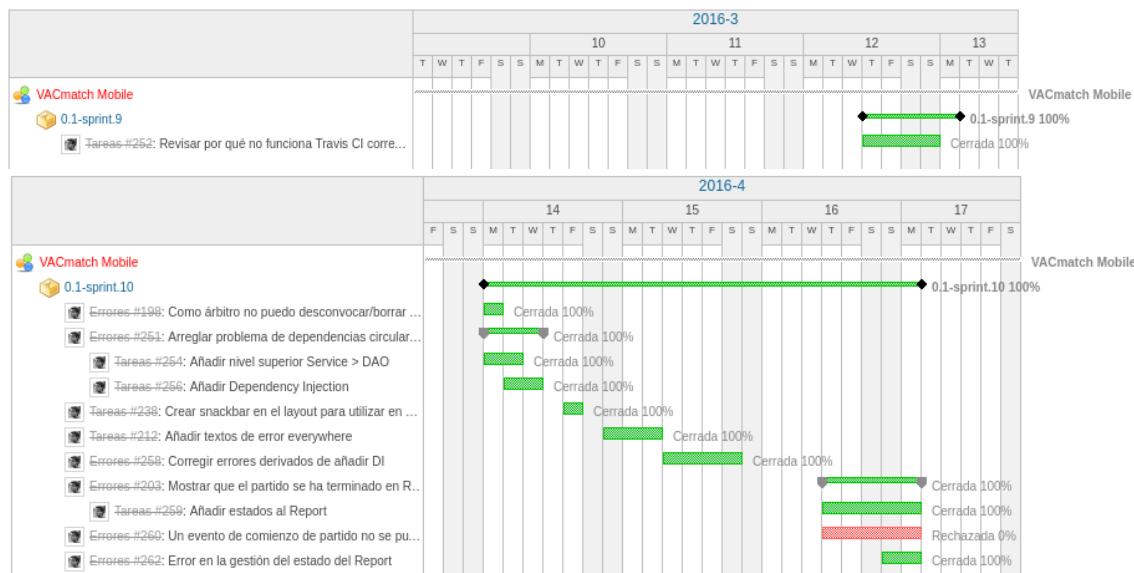


Figura 5.10: Diagramas de Gant dos sprints 9 e 10

5.3.4. Release 0.2.0: Usabilidade en menús

5.3.4.1. Planificación e definición da iteración

Esta iteración transcorre entre o día 25 de Abril e o 9 de Maio, unha iteración cunha única tarefa pero de abondo tamaño para cubrir a totalidade do sprint, centrada en engadir os enlaces que faltaban no menú lateral esquierdo e no superior derecho, certos botóns de retroceso así como correxir diversos erros menores.

5.3.4.2. Concurso Universitario de Software Libre

Así mesmo, durante este iteración recibíuse o “Premio al mejor proyecto de tecnologías móviles” dentro do Concurso Universitario de Software Libre (CUSL), un concurso onde participaron máis de 75 estudiantes de toda España con un total de 47 proxectos de código libre.

Fomos invitados a participar na fase final que tivo lugar os días 5 e 6 de Maio na Universidade de Sevilla na que presentar o proxecto ante representantes de diversas empresas de software libre españolas que tamén participaron con diversas charlas sobre os seus modelos de negocio e as vantaxes do software libre a nivel empresarial.

Foi unha gran experiencia a compartida con tódolos finalistas e asistentes e, por suposto, os custes da viaxe foron subvencionados pola organización e o premio finalmente foi de 300 € en metálico.



Figura 5.11: Finalistas CUSL.

5.3.4.3. Revisión e feedback

Aproveitando a participación no CUSL obtivérónse múltiples consexos e suxerencias, sobre todo en temas concretos de desenvolvemento xa que o xurado realizou unha profunda revisión do código da aplicación.

Entre elas destaca sobre todo a proposta de tratar de cambiar os múltiples callbacks por Promises de Javascript así como outros consellos para facilitar o mantemento do código como evitar encadear demasiados métodos.

Paralelamente publicouse unha entrada no blog no que comentar os diversos problemas xurdidos que obligaron a engadir inxección de dependencias así como a enorme refactorización de código levada a cabo para introducir tests automáticos e integración continua.

5.3.4.4. Tarefas e seguimento

Únicamente se realizou unha tarefa durante este período:

S0.2.0-1 Revisar os links nos menús e engadir información.

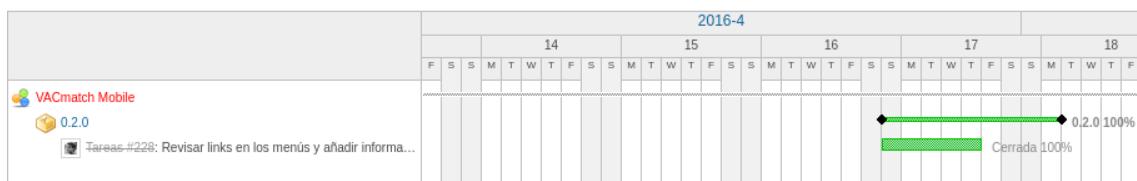


Figura 5.12: Diagramas de Gant da release 0.2.0

Esta tarefa tivo unha planificación moito máis reducida do habitual (16 horas) xa que ao resultar finalistas no CUSL, debeuse viaxar a Sevilla durante un total de tres días.

Finalmente o tempo invertido no desenvolvemento foi un total de 14 horas, aproveitando as 2 horas restantes para preparar a presentación para o concurso.

5.3.5. Release 0.2.1: I18n e app híbrida

5.3.5.1. Planificación e definición da iteración

Esta iteración comeza o día 9 e remata o 24 Maio.

A tarefa de maior tamaño que se planificou nesta iteración foi a internacionalización coa librería React Intl xa que supón modificar tódalas vistas da aplicación.

Posteriormente engadíuse Apache Cordova para permitir crear aplicacíons híbridas que funcionen en múltiples sistemas operativos móveis e por suposto, tamén se incluíu no repositorio de código, a documentación sobre cómo arrancar unha base de datos CouchDB para utilizar como backend e sobre cómo realizar a compilación da aplicación para executar nun sistema operativo móvil.

5.3.5.2. Revisión e feedback

Durante esta etapa tampouco se recibiu feedback nin se escribiu ningunha entrada no blog.

5.3.5.3. Tarefas e seguimento

S0.2.1-1 Engadir internacionalización.

S0.2.1-2 Crear app híbrida con Apache Córdova.

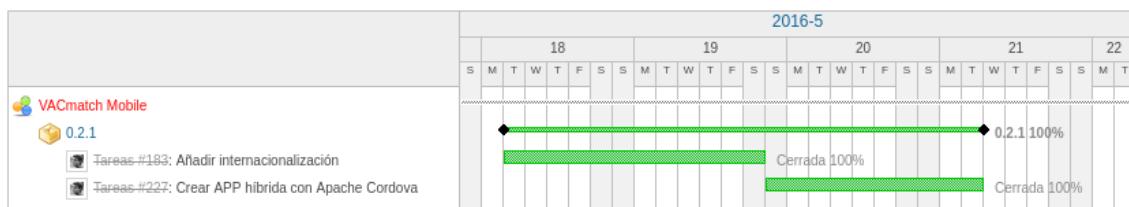


Figura 5.13: Diagramas de Gant da release 0.2.1

A duración estimada da iteración foi de 36 horas e finalmente realizouse en 1 hora más do previsto que foi imputada como hora extra para non retrasar o desenvolvemento.

5.3.6. Release 0.2.2: Finalización da memoria e revisión de erros

5.3.6.1. Planificación e definición da iteración

Esta última iteración comeza o 25 de Maio e exténdese finalmente ata o día 19 de xuño.

Céntrase principalmente en engadir certas vistas da aplicación como a de "Acerca de" ou a de configuración así como múltiples detalles visuais como engadir unha transición mentres os datos non se atopan cargados e solucionar diversos erros atopados na aplicación.

Así mesmo tamén se adicou un tempo considerable á finalización da memoria.

5.3.6.2. Revisión e feedback

Durante esta iteración coñeceuse que VACmatch foi invitado a participar na fase final do "Certamen de Proyectos Libres de la Universidad de Granada" co presente proxecto pero non foi posible asistir polo que se decidiu enviar un vídeo para realizar a presentación e polo tanto tampouco se puido obter feedback.

5.3.6.3. Tarefas e seguimento

Ao longo da iteración realizáronse as seguintes tarefas:

S0.2.2-1 Error ao engadir un evento, cámbiase o nome dos equipos

S0.2.2-2 Error ao tratar de asinar un acta como árbitro e como xogador creado na app móvil

S0.2.2-3 Eliminar páxina de home

S0.2.2-4 Non se engaden ao menú lateral esquierdo en ningún momento a lista de actas

S0.2.2-5 Un xogador debe poder asinar un acta incluso si é creado manualmente na acta

S0.2.2-6 Os links do menú lateral esquierdo non funcionan

S0.2.2-7 [Memoria] Apendices

S0.2.2-8 [Memoria] Conclusíons

S0.2.2-9 [Memoria] Deseño e implementación

S0.2.2-10 [Memoria] Planificación e seguimento

S0.2.2-11 [Memoria] Revisión xeral

S0.2.2-12 Só mostrar/editar as actas do usuario que está logueado

A estimación inicial para a realización desta iteración foi dun total de 91 horas, cumpríndose correctamente e dispoñendo finalmente de 10,5 horas libres para adicar a aprender tecnoloxías

que é posible que se utilicen nun futuro cercano como é o caso de Docker, para realizar entregas contínuas.

Na Figura 5.14 pódese observar o Diagrama de Gant da iteración.

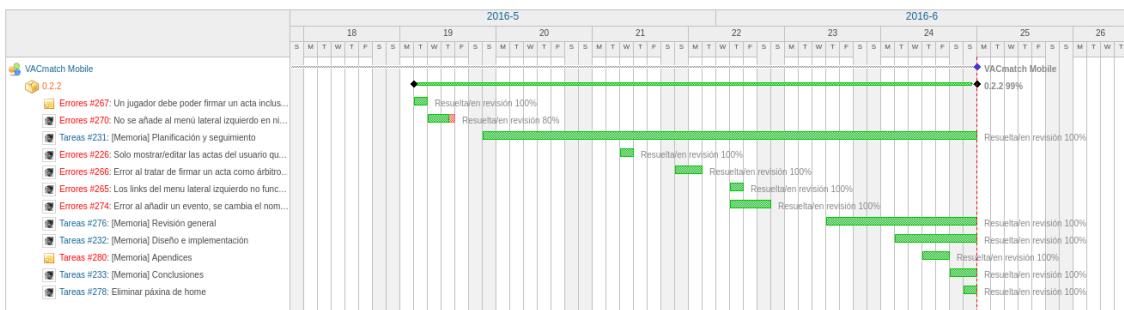


Figura 5.14: Diagramas de Gant da release 0.2.2

Capítulo 6

Fundamentos tecnolóxicos

Índice xeral

6.1. Linguaxes e frameworks empregados	46
6.2. Bases de datos	47
6.3. Estándares de comunicación	47
6.4. Repositorios de código	48
6.5. Ferramentas de xestión	48
6.6. Ferramentas documentais	49

NESTE capítulo móstraranse as diversas tecnoloxías que foron empregadas durante o desenvolvemento do proxecto así como ferramentas de xestión, bases de datos, repositorios de código e ferramentas documentais, todas, tecnoloxías Software Libre.

6.1. Linguaxes e frameworks empregados

ReactJS facebook.github.io/react

React é unha librería de Javascript para a creación de Single Page Applications (SPAs) [MP], permitindo crear aplicacións completas que se executen no navegador de forma sinxela a través de diversos compoñentes que se agrupan e que permiten crear aplicacións multiplataforma.

Reflux github.com/reflux/refluxjs

É unha implementación da arquitectura Flux que explicaremos no Capítulo 7.1 e que permite un fluxo de datos unidireccional, en lugar do bidireccional que é habitual nas aplicacións web. Permite tamén a creación de Stores nas que se mantén o estado das aplicacións e que permite compartir dito estado entre os diversos compoñentes da aplicación.

Jest facebook.github.io/jest

É unha ferramente sinxela creada sobre o framework de testing para Jasvascript, Jasmine, que facilita a utilización de mocks e a creación de tests unitarios.

React Intl (i18n) formatjs.io

React Intl é unha ferramenta para facilitar a internacionalización de aplicacións Javascript e concretamente as aplicacións baseadas en React.

React Router github.com/reactjs/react-router

Unha librería para o enrutado de aplicacións baseadas en ReactJS proveendo unha API sinxela con funcionalidades de gran potencia como a carga preguiceira de código ou o enrutado dinámico.

Material UI material-ui.com

Un conxunto de componentes para React que implementan o Material Design impulsado por Google, unha nova linguaxe visual baseada na representación en 3D dos obxectos que non deben intersecarse se non que a través de sombras para simular diferentes profundidades, os obxectos debe superpoñerse uns sobre os outros.

6.2. Bases de datos

CouchDB couchdb.apache.org É unha base de datos pensada para web que permite almacenar os datos en formato JSON e acceder aos mesmos a través dun navegador via HTTP, funcionando como unha API REST¹.

Permite múltiples funcionalidades pouco habituais entre os sistemas de xestión de bases de datos como servir aplicacións directamente dende CouchDB así como un sistema de replicación incremental e de detección de conflictos.

PouchDB pouchdb.com

Unha base de datos NoSQL baseada en Javascript e inspirada en CouchDB, pensada para facilitar o funcionamento de aplicacións web de forma offline.

PouchDB permite almacenar os datos localmente no navegador web cando non hay conexión a internet e sincronizar de forma sinxela ditos datos en remoto con CouchDB e outros servidores compatibles.

6.3. Estándares de comunicación

JSON couchdb.apache.org

É un formato estándar para o intercambio de datos e que pola súa simplicidade estase a impoñer como formato habitual por exemplo, para a comunicación con APIs REST e debido a súa similitude coa definición de obxectos en Javascript, permite que sexa tremadamente sinxelo traballar con él dende esta linguaxe.

¹Representational state transfer

6.4. Repositorios de código

Gerrit gerritcodereview.com

Gerrit é un repositorio de código baseado no sistema de control de versións Git e centrado en proveer un xeito sinxelo de realizar revisións de código dende unha plataforma web. Foi utilizado durante o desenvolvemento do prototipo da aplicación pero finalmente decidíuse trasladar o código a GitHub para facilitar a colaboración de outros posibles desenvolvedores.

GitHub github.com

GitHub é un repositorio de código que se está a convertir no lugar máis importante de publicación de aplicacóns Software Libre e que permite aloxar proxectos como o presente, de forma totalmente gratuita. Cómprase destacar que esta é a única ferramenta utilizada para o desenvolvemento do proxecto que non é software libre pero si proporciona unha visibilidade de cara a comunidade de gran importancia neste tipo de proxectos.

6.5. Ferramentas de xestión

Git git-scm.com

É un sistema de control de versións software libre con grandes funcionalidades e que é utilizado en millóns de proxectos. Aporta unha versatilidade enorme ao ser distribuído, permitindo traballar incluso de forma offline.

Gulp gulpjs.com

Un sistema que permite a automatización de tarefas durante o desenvolvemento de aplicacóns como por exemplo compilar automáticamente o código Javascript escrito na súa última versión á versión máis antiga para que poida ser executada por calquera navegador web.

Babel babeljs.io

É un compilador de Javascript que permite a traducir código fonte escrito no estándar ECMAScript 6 a ECMAScript 2015, soportado pola gran maioría de navegadores.

Browserify browserify.org

É unha ferramenta que permite escribir os módulos da aplicación como se fosen módulos para unha aplicación escrita en Node.js e que os compila para poder ser utilizados no navegador web.

Redmine redmine.org

É unha ferramenta de xestión de proxectos flexible, multiplataforma e software libre con diversos plugins para facilitar a planificación de iteracións e traballar con metodoloxías áxiles de desenvolvemento.

Travis CI travis-ci.org

É unha ferramenta de integración continua que permite automatizar a execución de tests ou o despregamento automático de código. Ademáis dispón dunha integración con Github polo que resulta moi sinxelo automatizar estas tarefas.

Docker [docker.com](https://www.docker.com)

É un sistema que permite empaquetar e despregar de xeito sinxelo aplicacións coas súas dependencias en unidades estándar chamadas contenedores, abstraendo e automatizando a virtualización da plataforma na que correrá a aplicación.

Atom atom.io

Un editor de texto software libre deseñado inicialmente por GitHub e centrado na súa extensibilidade gracias a un sinxelo sistema de plugins. Ademais, no ecosistema de ReactJS, existen compoñentes para Atom co obxectivo de facilitar a edición de aplicacións React, elaboradas polos mesmos impulsores da propia librería.

Apache Cordova cordova.apache.org

Unha ferramenta de desenvolvemento que permite usar tecnoloxías web estándar (HTML, CSS3 e Javascript) para crear aplicacións móveis multiplataforma.

6.6. Ferramentas documentais

LaTeX [latex-project.org](https://www.latex-project.org)

Un sistema para a composición de documentos que inclúe todo tipo de funcionalidades para a edición de textos científicos ou técnicos, moi adecuado para este proxecto e que xenera documentos de xeito sinxelo e automáticamente estruturados.

Dia [dia-installer.de](https://www.dia-installer.de)

É unha aplicación para a creación de diagramas entre os que se atopan os diagramas UML e que permite a exportación dos mesmos a imaxes vectoriais.

Capítulo 7

Deseño e implementación

Índice xeral

7.1. ReactJS e Flux	51
7.1.1. Introducción e elección da tecnoloxía	51
7.1.2. Elementos básicos	52
7.1.3. Estrutura da aplicación	54
7.2. Bases de datos e funcionamiento offline	55
7.2.1. PouchDB	56
7.2.2. CouchDB	56
7.2.3. Sincronización e xestión de conflictos	56
7.3. App híbrida con Apache Cordova	56
7.4. Interface gráfica e usabilidade	58
7.4.1. Elementos comúns	58
7.4.2. Iniciar sesión	62
7.4.3. Listado de actas	63
7.4.4. Acta	63
7.4.5. Finalización do encontro	66
7.4.6. Modificación de estilos	68
7.5. Multideporte	69
7.5.1. Deporte	69
7.5.2. Roles de usuarios	70
7.5.3. Eventos	70
7.6. Deseño da DB	72
7.7. I18n	73
7.8. Inxección de dependencias	73
7.9. Testing e Integración continua	74
7.9.1. TDD e BDD	75
7.9.2. Jest	75
7.9.3. Travis CI	78

NESTE capítulo verémolos detalles de deseño e implementación de diversas partes do proxecto e falaremos das decisións tomadas con respecto a utilización das múltiples tecnoloxías utilizadas.

Comezaremos facendo unha pequena introdución á terminoloxía e ás tecnoloxías utilizadas no proxecto na Sección 7.1, xa que resulta fundamental para comprender as diversas explicacións técnicas sobre o deseño e a implementación.

Falaremos tamén das bases de datos non relacionais na Sección 7.2, comentaremos a necesidade da creación da aplicación como App híbrida e todo isto seguido dunha extensa análise sobre a usabilidade e as múltiples decisións que se tomaron ao respecto, debido a gran importancia que ten este apartado dentro do proxecto.

Por último tamén comentaremos algúns conflitos que xurdiu durante o desenvolvemento a nivel de dependencias entre paquetes ou testing e as decisións que se tomaron para solventalos.

7.1. ReactJS e Flux

7.1.1. Introdución e elección da tecnoloxía

Actualmente o mundo das aplicacións móbiles está en pleno crecemento e cada vez é máis sinxelo ver cómo pequenos comercios ou incluso eventos de ocio como festivais de música ou conferencias, teñen a súa propia aplicación móvil¹.

A sociedade está a eliminar un soporte tradicional como é o papel en tódolos aspectos da vida, dende a publicidade ata a propia xestión do traballo, e todo estase a dixitalizar, facilitando o traballo humano e reducindo os custes a medio prazo.

Neste contexto decidíuse optar por realizar unha aplicación móvil híbrida, con tecnoloxías web, xa que se pode observar un movemento nos últimos anos cara este tipo de tecnoloxías que permiten desenvolver únicamente unha aplicación e executala nos diversos sistemas operativos móbiles existentes, sen ter que desenvolver unha aplicación concreta para cada un de eles.

Tras unha análise das diversas linguaxes e frameworks que podían ser utilizados para este fin, finalmente optouse por utilizar ReactJS pola sua flexibilidade fronte a outras alternativas como AngularJS ou EmberJS.

Estas alternativas céntranse en ofrecer un framework moi completo buscando cubrir tódolos aspectos dunha aplicación como o enrutado de urls, a internacionalización ou os servizos, fronte a React que únicamente trata de cubrir a xestión das vistas e do seu estado, dando total liberdade para escoller a tecnoloxía que se precise para o resto de compoñentes da aplicación.

Así mesmo tamén cómpre destacar a existencia de React Native [Eis], unha tecnoloxía baseada en React que permite crear aplicacións móbiles con tecnoloxías web e cunha experiencia de usuario

¹Eventos como o FOSDEM, a LibreCon ou OpenExpo xa disponen de aplicacións móbiles para facilitar que os asistentes organicen as actividades ás que asistirán.

exactamente igual a unha aplicación nativa tradicional.

Unha vez seleccioado React, optamos por seguir a arquitectura máis habitual dentro deste tipo de aplicacións, Flux, a través da librería Reflux.

7.1.2. Elementos básicos

7.1.2.1. Compoñentes de React

React é unha librería de Javascript que nos permite xestionar de xeito sinxelo as vistas da nosa aplicación a través de diversos elementos denominados “Compoñentes”.

React permítenos escribir os nosos compoñentes cunha sintaxis moi similar a HTML pero posteriormente traduce esta sintaxis a código javascript habitual polo que podemos decir que estamos escribindo páxinas web únicamente con funcións de javascript.

A idea principal é reutilizar e agrupar compoñentes para formar o que tradicionalmente chamamos vistax e que a súa vez poden ter un estado. Este estado pode variar modificando as vistax, como por exemplo cando se engade un novo gol na lista de eventos e React encárgase de volver a renderizar únicamente a parte da vista que cambiou, polo que é tremendaente eficiente.

7.1.2.2. Arquitectura Flux

Flux [Bod16] é unha arquitectura que básicamente propón o esquema mostrado na Figura 7.1.

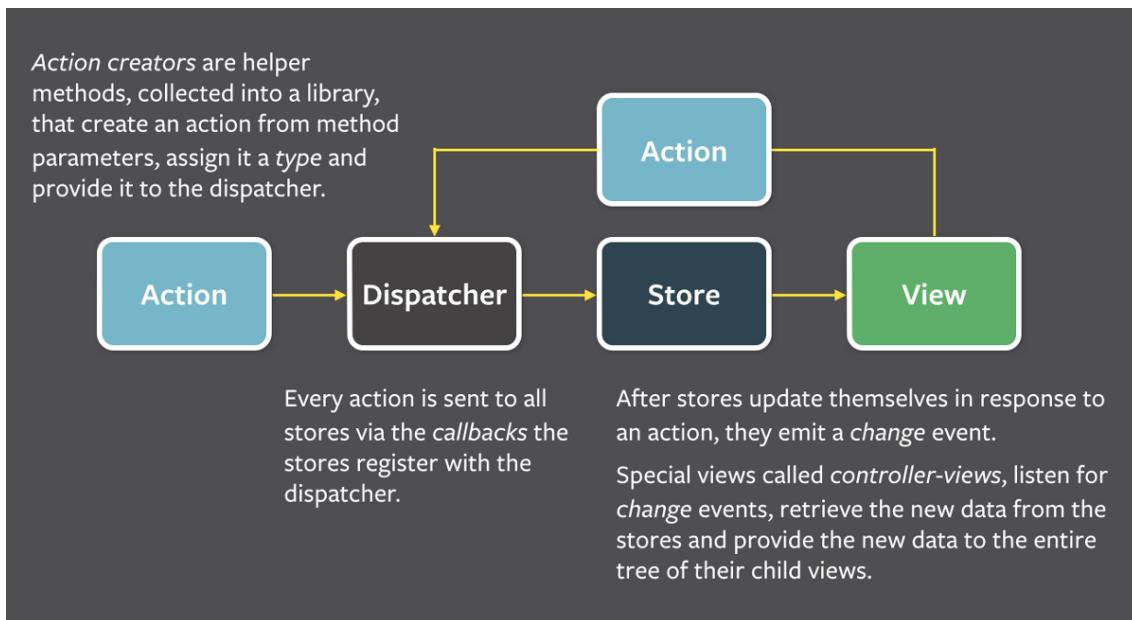


Figura 7.1: Esquema da arquitectura Flux

Unha aplicación que opte por seguir esta arquitectura debe conter os seguintes compoñentes:

7.1.2.2.1. Actions A vista atópase formada por unha serie de compoñentes de React que son capaces de disparar as Actions, por exemplo ao pulsar un botón, e que funcionan de xeito similar a eventos, notificando ao Dispatcher cando se produce a súa execución.

7.1.2.2.2. Dispatcher O Dispatcher é o encargado de recibir e enrutar as Actions disparadas cara as Stores.

7.1.2.2.3. Stores Estas Stores son as encargadas de xestionar o estado dos compoñentes e toda a lóxica necesaria para actualizar o estado co novo cambio.

Se por exemplo ao pulsar un botón debería cambiar un texto que está no compoñente, poderíamos ter este texto almacenado no estado dentro dunha Store e inicializado a un valor baleiro. Unha vez se fai click no botón, unha Action chamará a unha Store que é a que contén a información sobre cómo cambiar o texto do estado e React encargarase de volver a pintar o novo valor do estado, dentro do seu compoñente.

7.1.2.3. Implementación de Flux. Reflux.

Flux é unha arquitectura, unha especie de patrón de desenvolvemento pero existen diversas implementacións da mesma e neste proxecto decidímonos por utilizar Reflux².

O funcionamento é moi similar pero con algúns matices; por exemplo, non existe un único Dispatcher central que enruta as Actions se non que tódalas Stores están escuchando e reaccionan cando teñen un método para responder a Action.

Vexamos un exemplo práctico:

Listing 7.1: Exemplo de Action.

```
let TextActions = Reflux.createActions([
  "updateText"
])
```

Definiríamos unha Action (Fragmento de código 7.1), que queremos lanzar para tratar de actualizar o texto e crearíamos unha Store (Fragmento de código 7.2) que escucha as TextActions e que implemente unha función onUpdateText que será a que reciba o novo valor e actualizará o estado.

Listing 7.2: Exemplo de Store.

```
let TextStore = Reflux.createStore({
  listenables: TextActions,
  init: function () {
    this.state = ''
  },
  getInitialState: function () {
    return this.state
  },
})
```

²[Reflux](https://github.com/reflux/refluxjs) github.com/reflux/refluxjs

```
onUpdateText: function (newText) {
  this.state = newText
  this.trigger(this.state)
}
})
```

Por último definir o compoñente (Fragmento de código 7.3). Este deberá incluir unha lista de mixins onde indicarlle cales son as Stores que reaccionan ante eventos lanzados dende este compoñente, neste caso, a `TextStore`.

O compoñente tamén terá un método render no que definir o que poderíamos chamar informalmente a “vista” do que queremos renderizar e que inclúe un enlace para cambiar o texto mostrado.

Dito enlace reacciona cando se fai click e chama ao método `handleChangeTextClick` que lanza a Action para que a Store a intercepte, cambie o estado e React renderice de novo, únicamente o que cambiou.

Listing 7.3: Exemplo de compoñente de React.

```
let EndReport = React.createClass({
  mixins: [
    Reflux.connect(TextStore, 'text')
  ],
  handleChangeTextClick: function () {
    TextActions.updateText('Nuevo texto a renderizar')
  },
  render: function () {
    return (
      <p>{this.state}</p>
      <a onClick={this.handleChangeTextClick}>Change text</a>
    )
  }
})
```

7.1.3. Estrutura da aplicación

A continuación móstranse os diversos elementos xerais que contén a aplicación, as carpetas principais e certos ficheiros fundamentais na execución da aplicación.

tests Carpeta onde se poden atopar os tests da aplicación. De momento só se dispón de tests da capa de servizos.

i18n Nesta localización pódese atopar a internacionalización, coas cadeas dentro da subcarpeta de “messages” e coa descripción das mesmas para cada compoñente de React dentro de ”descriptors“, coa idea de darlle contexto e facilitarlle ao tradutor a súa labor.

cordova app Aquí podemos atopar a configuración para compilar a aplicación aos diversos dispositivos móbiles.

src/app Estrutura principal da aplicación

Actions Lista de ficheiros con accións a disparar, agrupados cada un pola store que o implementa.

API Contén a configuración, as urls da aplicación e a factoría utilizada para a inxección de dependencias dos servizos que explicaremos máis adiante.

Components Lugar onde se poden atopar tódolos componentes que forman as vistas da aplicación.

Daos Entidades que abstraen dos servicios a definición do acceso a base de datos, facilitando que sexa sinxelo cambiar dita base de datos se é preciso.

Models Contén a definición dos modelos da aplicación, tanto as entidades que se almacenan en base de datos como de certas clases que son utilizadas pola aplicación, como por exemplo as que definen os deportes ou a implementación dos tipos de eventos.

Services Estas clases conteñen a lóxica de negocio da aplicación.

Stores Lista de ficheiros que conteñen e xestionan o estado da aplicación. Cómpre diferenciar a lóxica de negocio, que se almacena nos Servizos, da xestión do estado dos componentes de React que podemos atopar nas Stores.

app.js Ficheiro de inicialización da aplicación que se encarga de arrancar o enrutador e executa a aplicación ben en modo web ou ben en modo aplicación móvil en función da configuración definida.

router.jsx É o encargado de poñer en relación as urls da aplicación cos componentes que corresponden a cada unha e onde se introduce tamén, a información sobre a internacionalización.

7.2. Bases de datos e funcionamento offline

O mundo das bases de datos está a cambiar enormemente nos últimos anos coa aparición das bases de datos non relacionais [Tiw11], neste caso concreto, era preciso dispoñer dunha base de datos no cliente que permitise almacenar os datos das actas dos encontros de forma offline xa que o árbitro do encontro, pode non ter cobertura e debería poder cubrir a súa acta da mesma forma.

Ante este requisito cabe pensar en bases de datos lixeiras como SQLite que se utilizan habitualmente nas aplicacíons móbiles pero ditas bases de datos non poden ser executadas directamente nun navegador web, polo que finalmente se optou por utilizar PouchDB, unha base de datos orientada a documentos e creada sobre o GlobalStorage³ do navegador, pensada dende o primeiro momento para crear aplicacíons web que funcionen de forma desconectada.

³Un obxecto que mantén múltiples áreas de almacenamento privado para almacenar datos durante un largo periodo de tempo no navegador web

7.2.1. PouchDB

As actas electrónicas son documentos que unha vez sexan creadas, apenas serán modificadas e neste tipo de información, son as bases de datos orientadas a documentos as que presentan un mellor rendemento.

A elección de PouchDB⁴ foi sobre todo motivada por ser tremendamente lixeira, multinavegador e sobre todo por facilitar a sincronización contra unha base de datos remota, neste caso, CouchDB.

7.2.2. CouchDB

CouchDB⁵ é a base de datos non relacional que se utiliza como base de datos central en remoto, e coa cal sincronizará PouchDB que se atopa no navegador web do cliente.

Esta base de datos é tremendamente versatil e sinxela de utilizar xa que implementa unha API REST e, a forma de interactuar con ela basease simplemente en enviar documentos JSON a través de sinxelas peticións HTTP cara a súa API.

Destaca pola súa extensa comunidade e por ser altamente dispoñible e tolerante a erros pero eventualmente inconsistente, aínda que foi crítico na súa elección a xestión que CouchDB fai dos conflictos de datos e que comentamos a continuación

7.2.3. Sincronización e xestión de conflictos

Probablemente o maior reto á hora de enfrentarse a unha aplicación con funcionamento offline é a xestión de conflictos entre os datos.

Neste caso en concreto, existe a problemática de que por múltiples motivos, un acta pode ser modificada en dous lugares ao mesmo tempo, tanto polo árbitro no seu teléfono como pola persoa encargada da xestión da federación polo que resulta imprescindible non perder información e ser capaces de mostrarlle a totalidade da información ao xestor da federación para que este poida seleccionar cales datos son os reais.

Neste punto é donde CouchDB resulta moi útil xa que a propia base de datos se encarga de almacenar unha árbore co histórico de revisións que se producen sobre un documento e así se poderían mostrar ao usuario para que seleccione a correcta en caso de conflicto.

7.3. App híbrida con Apache Cordova

O obxectivo do proxecto é poder utilitzalo en múltiples dispositivos móbiles co fin de facilitar que calquera árbitro poida utilizar a aplicación independentemente do sistema operativo que teña o seu teléfono ou tableta.

⁴PouchDB pouchdb.com

⁵CouchDB couchdb.apache.org

É por isto que é fundamental a utilización dun sistema que permita este desenvolvemento multiplataforma como é o caso de Apache Cordova⁶ que, ademáis, facilita o acceso aos elementos do dispositivo como sensores, datos, estado de rede... a través dunha serie de APIs estandar.

Actualmente non se está a utilizar ningunha de estas funcionalidades pero si é posible que nun futuro cercano se decidan implementar novas características que si precisen o acceso a este tipo de elementos como pode ser por exemplo, o micrófono, para que o árbitro poida gardar as incidencias dun partido como notas de voz en lugar de escribillas.

Na Figura 7.2 podemos ver un esquema da arquitectura dunha aplicación sobre Apache Córdova.

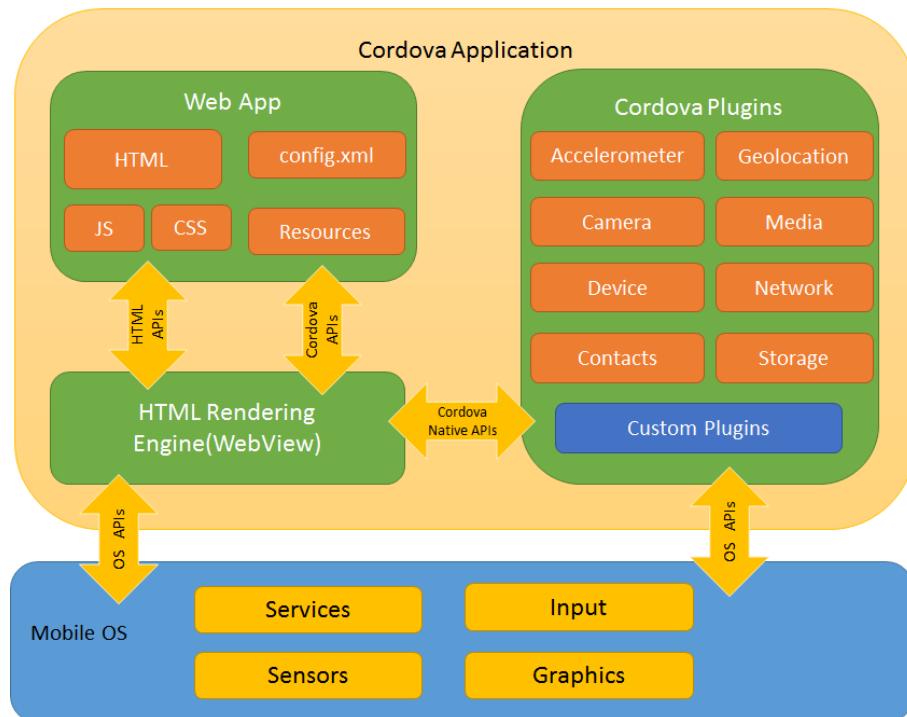


Figura 7.2: Arquitectura dunha aplicación con Apache Cordova

Básicamente a aplicación executase como unha aplicación web normal sobre un WebView, un motor de renderizado de HTML que pode interactuar coas APIs nativas do dispositivo a través dunha serie de plugins que Cordova provee.

Para executar a aplicación únicamente é preciso crear un proxecto de Cordova, engadir os ficheiros da aplicación, as plataformas para as que se deseñe xerar o proxecto e construílo.

⁶Apache Cordova cordova.apache.org

7.4. Interface gráfica e usabilidade

A importancia da interface gráfica en este proxecto é crucial, debido a que a aplicación será utilizada por persoas dun amplísimo rango de idades polo que a a usabilidade é o valor máis importante.

De feito realizouse un forte traballo de campo con usuarios reais da aplicación para facer fincapé en simplificar ao máximo o proceso de creación de un acta deportiva.

Finalmente é importante diferenciar 4 grandes partes da aplicación a nivel visual: inicio de sesión, listado de actas, acta e finalización de encontro, así como unha serie de elementos comúns que son reutilizados ao longo da aplicación e que se comentarán a continuación.

7.4.1. Elementos comúns

Para facilitar o desenvolvemento da aplicación creáronse varios compoñentes de React xenéricos que permiten realizar tarefas comúns a múltiples partes da aplicación e compoñentes de React.

7.4.1.1. Menú lateral esquierdo

É o menú principal da aplicación e que se mostra ao pulsar no botón superior esquierdo dende a maior parte de vistas.

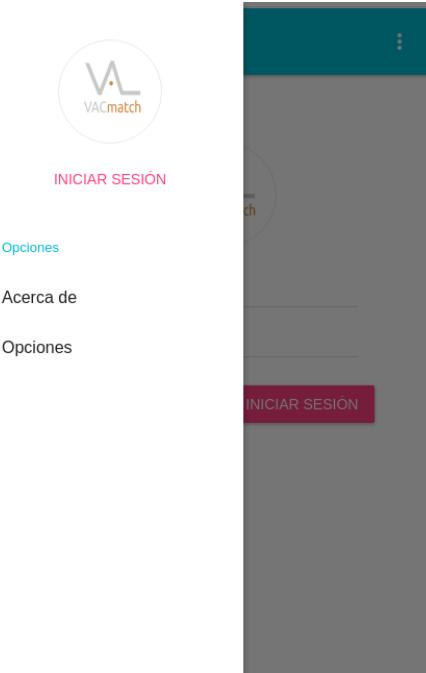


Figura 7.3: Menú lateral esquierdo.

Como se pode ver na Figura 7.3, este menú despregable mostra actualmente o logotipo de VACmatch e unha serie de enlaces entre os que se atopan as páxinas de configuración e a páxina

”acerca de“.

Para xestionar os enlaces que se mostran en cada páxina de forma internacionalizada, e para permitir dispoñer dos elementos nun sitio centralizado, creouse o compoñente `LeftMenuItems`. Este compoñente contén varias listas con items para introducir dentro do menú lateral esquierdo e que se poden editar ou engadir outras se é preciso.

Utilízase tamén a `MenuStore` para xestionar os elementos que se atopan actualmente no menú e pódese utilizar a Action `setLeftMenu` para modificar os elementos que o componen en calquera momento, por exemplo, ao entrar nunha nova vista.

7.4.1.2. Enlaces do menu superior dereito

O funcionamento deste menú é moi similar ao anterior só que este non dispón de compoñente propio e únicamente é preciso chamar a Action `setRightMenu` para colocar a listaxe cos novos elementos a mostrar, no menú despregable como se ve na Figura 7.4

A función debe recibir unha lista de obxectos que conterán os seguintes atributos:

text Texto a mostrar polo item.

url Url a onde ir ao facer click no elemento.

callback Función de callback que se executará unha vez se redirixa a aplicación á url indicada (Opcional).

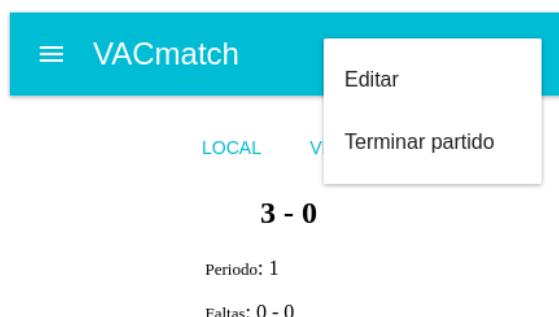


Figura 7.4: Menú despregable dereito.

7.4.1.3. Información e axustes

Toda aplicación debe conter un apartado de información sobre a mesma, e neste caso móstranse ademais as diversas redes sociais do proxecto e o repositorio de código en GitHub para que calquera poida contribuir ao mesmo.

Por outro lado tamén se ten en conta que toda aplicación debe ser personalizable en certa medida, de momento non se dispón de opcións máis ca cambiar de idioma pero cando exista a necesidade

de incorporar novos elementos de configuración, este será o apartado onde os usuarios poderán modificar ditas opcións.

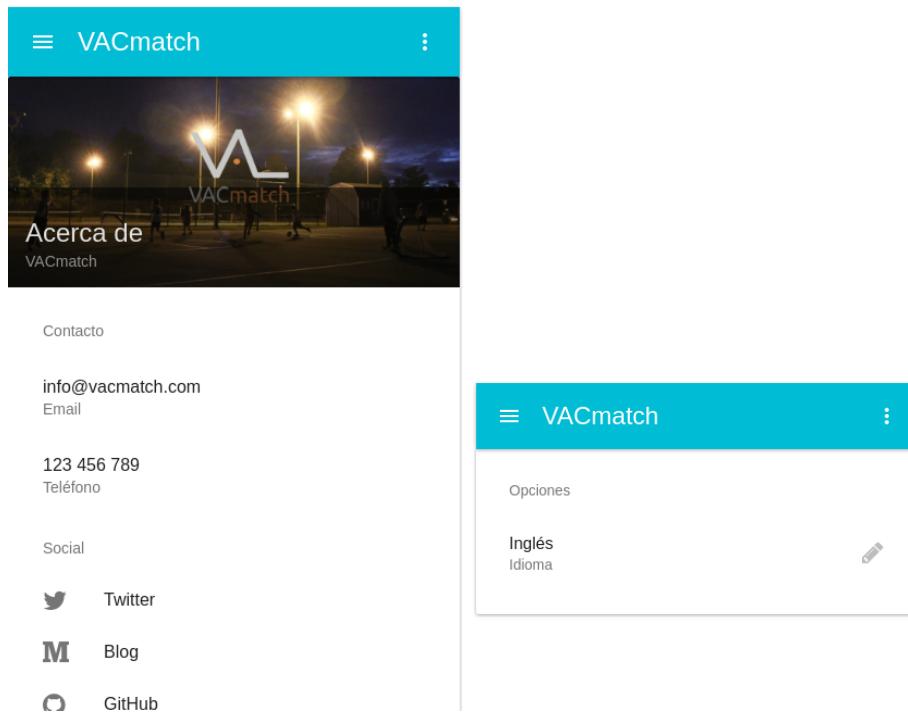


Figura 7.5: Ventás de información e axustes.

7.4.1.4. Barra de notificacións

Inicialmente xurdíu a necesidade de comunicarlle ao usuario os errores que se producen na aplicación, casos concretos como introducir un usuario incorrecto ou tratar de eliminar un evento de comezo de partido antes de eliminar o de fin do mesmo, deben mostrar un aviso ao usuario.

É por iso que se decidíu crear unha “Barra de notificacións” (Figura 7.6), unha pequena ventá que surde a modo de aviso na parte inferior da pantalla durante uns segundos para mostrar información.

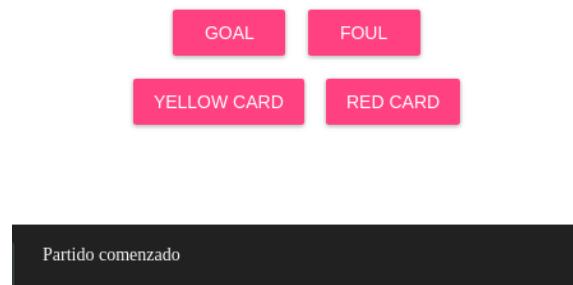


Figura 7.6: Barra de notificacións.

Inicialmente pensouse para mostrar os errores pero tamén resulta de utilidade a hora de mostrarlle

outra información ao usuario como cando un evento é engadido correctamente.

As novas mensaxes son almacenadas na SnackbarStore creada para o caso, que permite diferenciar as mensaxes de erro e as de información, facendo que a implementación da barra sexa moi sinxela como se mostra no Fragmento de código 7.4.

Listing 7.4: Barra de notificacións.

```
let SnackBarItem = React.createClass({
  mixins: [
    Reflux.connect(SnackBarStore, 'snackBar')
  ],
  onStatusChange: function (status) {
    this.refs.snack.show()
  },
  componentDidMount: function () {
    this.listenTo(SnackBarStore, this.onStatusChange)
  },
  render: function () {
    return <Snackbar key={'generic-snackbar'}
      ref='snack'
      message={this.state.snackBar.message}
      autoHideDuration={4000} />
  }
})
```

7.4.1.5. Autenticación

A autenticación é fundamental nesta aplicación xa que únicamente os usuarios con acceso poden crear ou modificar actas, neste contexto foi preciso crear un compoñente de autenticación que se encargue de realizar a comprobación de se o usuario iniciou sesión na aplicación sempre que se carguen os compoñentes.

Creouse unha clase sinxela que deben envolver a aqueles compoñentes que só poidan ser accedidos se o usuario está autenticado, o “AuthenticatedComponent” (Fragmento de código 7.5) e que redirixe a aplicación cara páxina de iniciar sesión, no caso de non poder atopar un usuario activo na sesión.

Listing 7.5: Compoñente de autenticación.

```
export default (ComposedComponent) => {
  let AuthenticatedComponent = React.createClass({
    mixins: [
      Reflux.connect(AuthStore, 'auth'),
      History
    ],
    componentWillMount: function (transition) {
```

```
        if (config._env !== 'development') {
            // This method is called before transitioning to this component.
            // If the user is not logged in, we'll send him or her to the Login page.
            if (!AuthStore.isLoggedIn()) {
                this.history.pushState(null, urls.login.show)
            }
        },
    }

    render: function () {
        return (
            <ComposedComponent
                {...this.props} />
        )
    }
})
return AuthenticatedComponent
}
```

Ao mesmo tempo, este compoñente permítenos aportar unha funcionalidade diferente en caso de atoparnos en modo “desenvolvemento”, saltando a comprobación de se o usuario ten sesión iniciada e permitíndonos acceder a calquera páxina sen ter que iniciar sesión cada vez que se recarga a páxina.

7.4.1.6. Lista de pestanas

Durante o desenvolvemento percatámonos de que é habitual a necesidade de utilizar unha lista de elementos divididos en varias pestanas, tanto á hora de mostrar a lista de xogadores dos equipos como a hora de asinar un acta ou engadir novos eventos.

É por isto que se creou un compoñente xenérico que permite incorporar unha lista de pestanas e unha lista de elementos para cada unha de elles, xerando de xeito sinxelo e xenérico estas vistas. Pódense introducir en elles calquera calquera tipo de elemento sempre que se encontre envolto dentro dun tipo xenérico definido como Item.

7.4.2. Iniciar sesión

Na Figura 7.7 podemos atopar a páxina inicial da aplicación que permite a un usuario iniciar sesión cos seus datos de acceso así como, pulsando o botón de rexistro, crear un novo usuario dentro da aplicación que lle permita xestionar actas de xeito manual.

Entre os múltiples parámetros que o usuario pode cubrir, atópase un código PIN que lle permitirá asinar as actas dos encontros que arbitre.

Cómpre mencionar que, á parte das páxinas de configuración da aplicación e da páxina de “acerca de”, esta é a única vista accesible para usuarios sen autenticar e calquera intento de acceso a algúns do resto, redirecciónará ao usuario cara esta páxina de inicio de sesión.

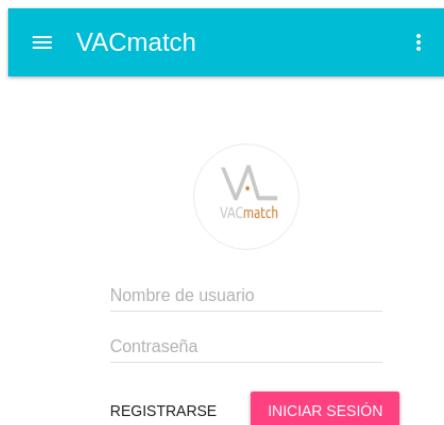


Figura 7.7: Vista de inicio de sesión.

7.4.3. Listado de actas

Nesta sección podemos ver a lista de actas que un árbitro ten asignadas na Figura 7.8, divididas en dúas pestanas, por un lado as de próximos partidos a arbitrar e por outro aquellas nas que o encontro xa rematou e que non é habitual que sexan accedidas.

Cada un dos elementos permítenos modificar a información básica da acta, tanto o nome dos equipos como o lugar de celebración ou a data por se é preciso edita ditos datos, permitindo do mesmo xeito eliminar a acta e tódolos elementos que a compoñen.

Ademais, esta páxina danos a opción de engadir unha nova acta se é preciso, unha funcionalidade imprescindible xa que por múltiples razóns poderíase dar o caso de que a aplicación dun árbitro non sincronice as súas actas coas existentes na base de datos da federación, polo que o árbitro debe poder crear un acta baleira de xeito manual incluso sen ter conexión a internet.

7.4.4. Acta

Esta é a parte central da aplicación, con total seguridade será o lugar onde os usuarios pasarán a gran maioría do seu tempo de uso da aplicación xa que é o punto central do desenvolvemento dun partido.

A acta dun encontro deportivo é un elemento con gran cantidade de información e que pode ser visualizada sen apenas esforzo xa que se atopa toda representada únicamente nunha folla de papel.

É por iso que dende o primeiro momento se tiña claro que había que tratar de manter esa

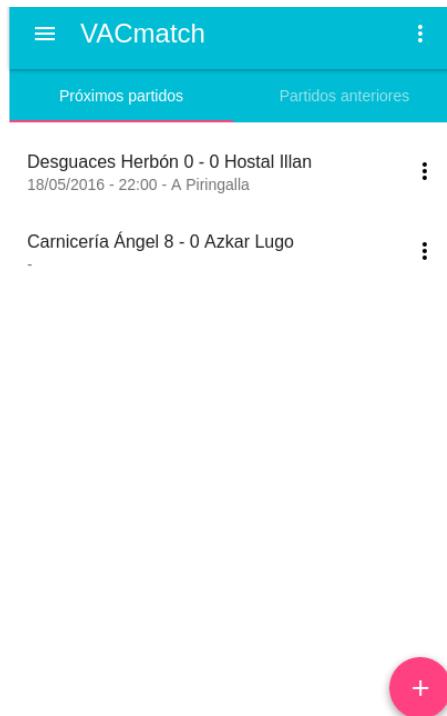


Figura 7.8: Listado de actas.

sinxeleza de uso pero, evidentemente, era imposible acceder a toda esa información en un só golpe de vista dentro dun soporte dun tamaño tan pequeno como é un teléfono móvil.

Entón o enfoque foi un pouco diferente e tratouse de facer que o proceso de cubrir un acta resultase o máis interactivo posible, mostrando na pantalla central únicamente a información indispensable para o árbitro que está a cubrila.

O deseño final pódese ver na Figura 7.9 na que se mostran os seguintes campos:

- Os nomes dos equipos que están a competir con un enlace para ver os seus xogadores.
- Dous campos de resultados.
- Un cronómetro.
- Un botón para comezar o encontro e xestionar o cronómetro unha vez comezou.
- Un botón de edición
- Un botón para mostrar os eventos que ocorreron.
- Botóns para engadir un dos diversos eventos que poden ser utilizados para este encontro



Figura 7.9: Vista principal dun acta.

7.4.4.1. Convocar xogadores

Nas accións habituais que debe realizar un árbitro, o primeiro paso é cubrir no papel os nomes dos xogadores que se atopan presentes no encontro coas fichas identificativas que cada un de eles aporta.

Como o obxectivo é eliminar o papel e simplificar o traballo do árbitro e da federación, a acta do encontro contén tódolos xogadores que se atopan inscritos no equipo correspondente coa sua foto como se pode ver na Figura 7.10, polo que xa non son precisas as fichas en papel de cada un de eles e o árbitro non ten que escribir os nomes dos integrantes dos equipos, se non que únicamente debe seleccionar cales de eles se atopan no campo.

Tamén ten a posibilidade de editar o dorsal do xogador para este encontro en concreto xa que en algunas competicións é habitual que os xogadores cambien de camiseta segundo o partido.

Por último existe a opción de engadir un novo xogador de xeito manual, útil para o caso no que o árbitro teña que crear un acta manualmente ou para cando un novo xogador é engadido a un club pero non é dado de alta na aplicación a tempo, e se o árbitro accede, este pode ser engadido a acta de xeito manual e competir sen problemas.

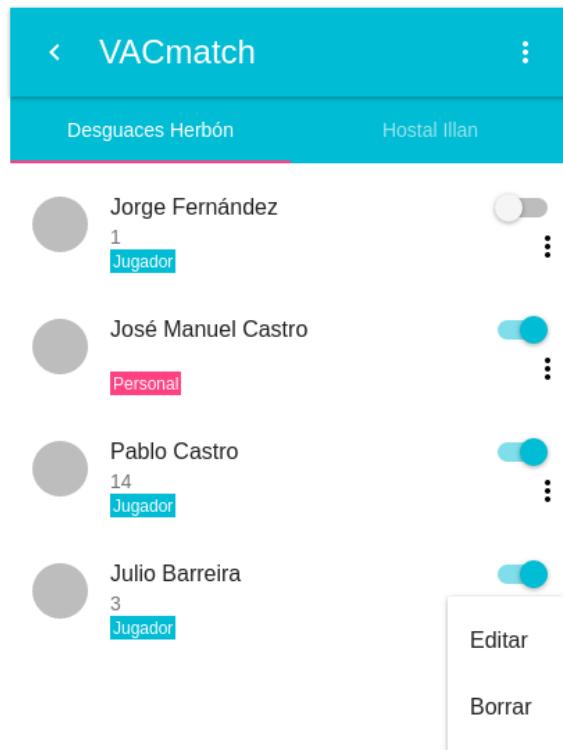


Figura 7.10: Seleccionar xogadores presentes no encontro.

7.4.4.2. Inicio e fin do partido

O seguinte paso á hora de xestionar a acta dun encontro é indicar cando comeza o mesmo, para isto o usuario dispón dun botón que só se mostra cando non comezou aínda dito partido e que se encarga de introducir un evento de comezo.

No momento en que se indique o comezo de partido, atoparase dispoñible o botón que permite arrancar e deter o **cronómetro**, co fin de poder xestionar o encontro de xeito interactivo e que tódolos eventos introducidos leven incorporado o momento no que se produciron.

Non é obligatorio utilizar o cronómetro polo que os eventos poden ser engadidos en calquera momento e non se almacenará o minuto no que se produciron.

Do mesmo xeito, pódese engadir un evento de fin de encontro nunha das opcións despregables do menú superior dereito e que redirixirá ao árbitro cara a páxina de asinado de acta.

7.4.5. Finalización do encontro

Cando un árbitro decide rematar un encontro, accederá a esta última vista da aplicación que podemos ver na Figura 7.11 onde poderá engadir un texto coa información de posibles **incidencias** que ocorreran durante o encontro e que deben ser postas en coñecemento da federación, como algúin

tipo de agresión ou o motivo polo que un partido tivo que ser finalizado antes de tempo.



Figura 7.11: Fin de encontro.

Dende o menú superior dereito pódese acceder directamente aos eventos que ocorreron durante o partido. Isto resulta útil para mostrarlle as persoas que asinan a acta toda a información da acta para que poidan confirmar que realmente os eventos gardados na mesma, son certos.

A vista ten varias pestanas, unha para o árbitro e outra para cada un dos equipos, onde poderán asinar a acta tantas persoas como sexa preciso.

Unha vez se faga click no botón de asinar de algunha das pestanas, mostrárase unha listaxe coas persoas que poden asinar de cada equipo ou do equipo arbitral como podemos ver na Figura 7.12.

Para realizar a sinatura, a persoa debe dispoñer de conta creada na base de datos da federación na que debeu de indicar previamente código PIN que lle permitirá asinar a acta.

No caso de xogadores creados manualmente dende a propia aplicación móvil, poderase asinar a acta sen dispoñer de código PIN, deixando o oco baleiro.

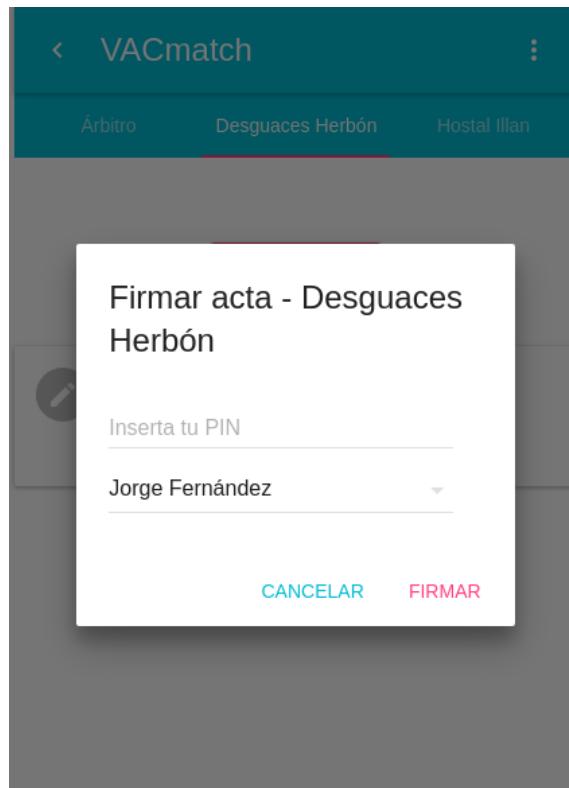


Figura 7.12: Sinatura dun acta.

7.4.6. Modificación de estilos

A xestión de estilos na aplicación varía de xeito considerable fronte ás aplicacións web habituais polo que é importante facer unha pequena reseña.

Certa parte da comunidade de React defende que o estilo das aplicacións web, que tradicionalmente se xestionava a través de follas CSS, debe mudar e pasar a realizarse directamente en Javascript.

Os estilos son inxectados directamente no compoñente de xeito “inline” a través do atributo “style” de HTML⁷, coa idea de acabar con diversos errores de deseño en CSS, algúns dos cales se comentan a continuación.

O feito de que os estilos sexan código global accesible dende calquera parte das aplicacións tradicionais, fai que esta sexa pouco consistente ante os cambios, unha pequena modificación pode cambiar gran parte da aplicación e pode ser difícil de atopar dito cambio cando os proxectos adquieren un certo tamaño polo que a xestión de xeito máis localizado a través de variables Javascript, facilitaría este mantemento.

A eliminación de código morto tamén é outra das vantaxes de utilizar Javascript para este

⁷A diferenza da maneira tradicional na que os estilos se componen en clases CSS a través do atributo ”class” e pódense anidar unhas con outras de forma declarativa

propósito xa que, agora os estilos serán variables locales que os “minifiers”⁸ de Javascript eliminarán automáticamente.

Por último tamén aporta unha total flexibilidade, dando incluso a posibilidade de tratar os estilos como parte do estado da aplicación e convertilos de este xeito, en totalmente dinámicos.

7.5. Multideporte

Un dos obxectivos iniciais era conseguir que a aplicación poidera adaptarse de xeito moi sinxelo a novos deportes, inicialmente comezaríase co fútbol pero todo o deseño debía estar orientado cara esta finalidade.

7.5.1. Deporte

Para iso creouse unha `SportStore` que contén a información do deporte que a aplicación está xestionando actualmente, e que pode ser actualizado e cambiado por outro, de xeito moi sinxelo.

Dende múltiples compoñentes como por exemplo o `PersonList`, pódese obter o deporte actual que está almacenado na Store e as múltiples propiedades que o definen, como se pode ver na Figura 7.13.

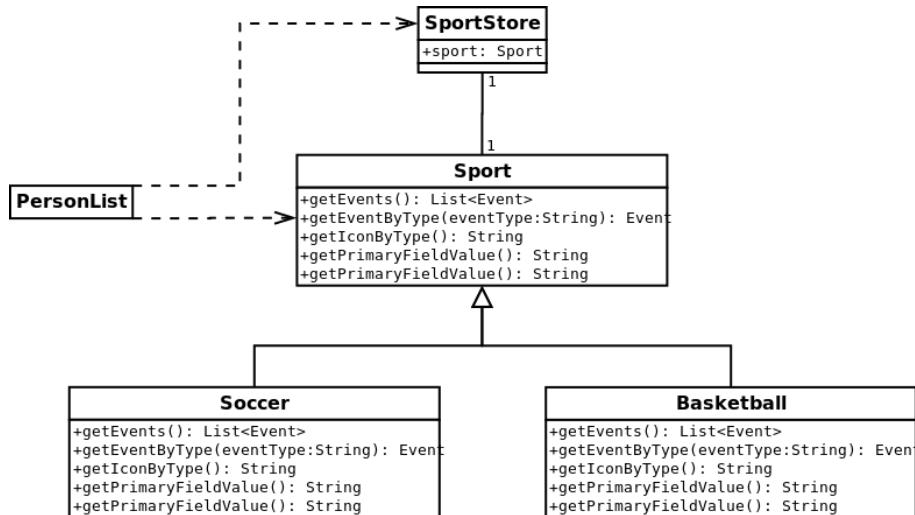


Figura 7.13: Patrón estratexia para o deporte.

Cada deporte debe extender a unha clase xenérica `Sport` nun patrón Strategy [GHJV94] que se pode observar na Figura 7.13 e que define unha serie de métodos que tódolos deportes deben implementar, métodos que indican cómo se realizan certas accións sobre os datos en cada deporte.

A continuación móstranse os parámetros que ten un `sport` actualmente:

⁸Programas que eliminan código e caracteres inecesarios como os espacios ou tabulacións

Eventos A lista de eventos que se poden realizar en ese deporte.

Evento por tipo Permite recuperar un obxecto Evento a partir da clave do seu tipo.

Icono por tipo Devolve o icono que identifica a un Evento dun tipo para renderizalo nalgúnha parte da aplicación como a lista de eventos.

Campo primario É unha función que calcula o resultado para almacenar no campo primario da acta en función da lista de eventos para ese encontro e de cada deporte. En fútbol devolve únicamente a suma de eventos de tipo gol.

Campo secundario Similar ao campo anterior pero para introducir información no campo secundario da acta, no caso do fútbol, o campo secundario almacena as faltas do encontro e esta función devolve a suma de faltas que hay no partido.

Esta implementación fai que sexa tremadamente sinxelo engadir un novo deporte, básicamente deberíase crear unha nova clase que extendese os métodos correspondentes coa funcionalidade concreta de ese deporte.

7.5.2. Roles de usuarios

Actualmente a aplicación únicamente pode ser accedida por árbitros, usuarios con un rol concreto, pero resulta trivial engadir novos roles, como por exemplo o de administrador dunha federación, para que este poida crear tamén encontros dende a aplicación móvil.

7.5.3. Eventos

Do mesmo xeito, cada deporte ten os seus propios eventos polo que se definíu unha forma de engadir eventos novos de xeito sinxelo, co fin de que a incorporación dun novo deporte resultase o más trivial posible.

Actualmente podemos dividir os eventos en dous tipos:

De control Son eventos xenéricos de xestión como cambiar de parte, ou comezar un encontro, que non é habitual que vaian a cambiar xa que son compartidos pola inmensa maioría de deportes.

De deporte Son eventos concretos de cada deporte como engadir un gol ou unha tarxeta, a súa vez divídense en:

Evento É un tipo de evento de deporte que únicamente permite ser engadido ou eliminado, non dispón dun comportamento especial.

Evento con causa Este tipo de evento, engade tamén unha causa pola que se producíu, útil para indicar por exemplo a motivación que levou a un árbitro a expulsar a un xogador con unha tarxeta vermella.

Evento de puntuación Tipo de evento que permite actualizar os campos de resultados, primario e secundario de forma automática ao ser engadido.

Na Figura 7.14 podemos ver a relación existente entre a `EventStore` que contén o estado dos eventos e os componentes da vista que os renderizan.

Temos dous componentes de React para renderizar os eventos dentro da vista que lista tódolos eventos que suceden durante o partido, os componentes `SportEvent` e `ControlEvent`.

Por outra banda temos tamén dous componentes de React que definen a vista que se ten que mostrar cando se deseñe engadir un evento de este tipo dende a aplicación, son `Event` e `EventWithCause` que ademais mostrará unha lista de causas que poden ser engadidas ao evento.

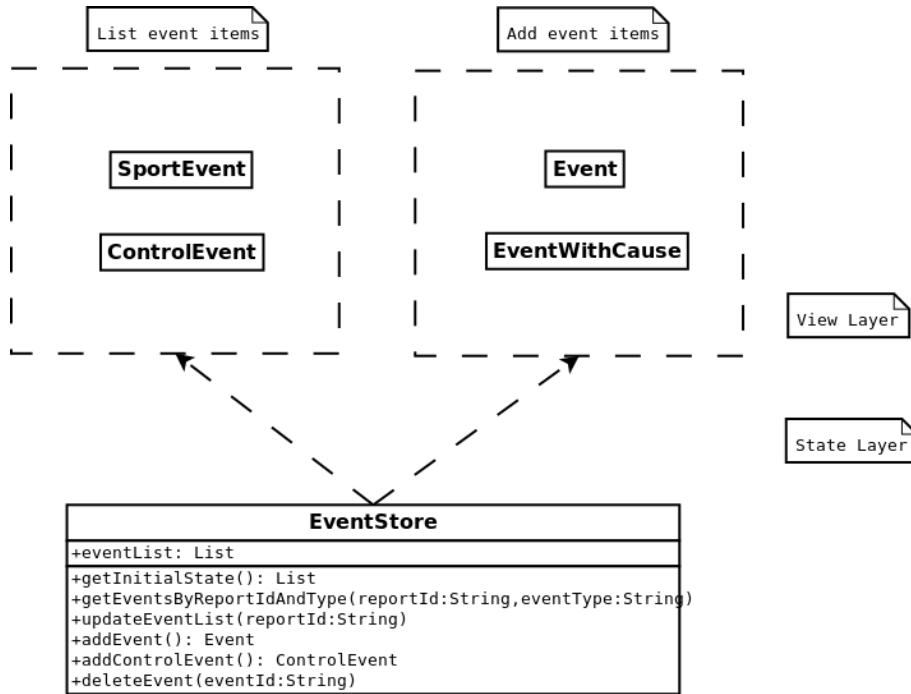


Figura 7.14: Diagrama parcial da vista dos eventos.

En definitiva, para crear un novo evento débese definir unha clase Javascript coas propiedades do mesmo e que implemente ao `SportEvent` como se pode ver na Figura 7.15 a través dos exemplos de `GoalEvent` ou `FoulEvent`.

Tamén é preciso engadir unha clase de React que defina as vistas que se mostrarán ao intentar engadir e listar un evento durante o encontro ou utilizar as xa definidas previamente que se poden ver na Figura 7.14.

Por último, como podemos observar de novo na Figura 7.15, débese engadir o evento creado anteriormente, dentro da lista de eventos que ten ese deporte, e que se atopan na clase que implementa dito deporte concreto.

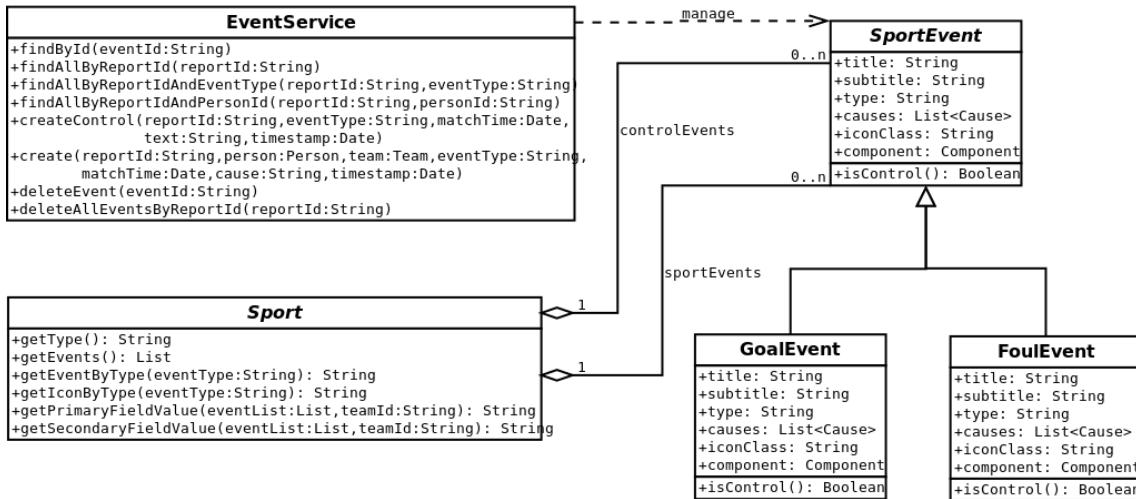


Figura 7.15: Diagrama de clases para a xestión de eventos deportivos.

7.6. Deseño da DB

VACmatch Mobile foi pensado inicialmente para ser integrado co sistema de xestión de competicións que estamos a desenvolver con VACmatch Web e que dispón dunha API de comunicacións que se atopa montada sobre unha base de datos relacional.

É por iso que o modelo de datos inicial estaba plantexado para unha base de datos relacional e tivo que ser adaptado para utilizar unha non relacional orientada a documentos como é PouchDB.

Prácticamente tódalas entidades foron parcialmente desnormalizadas para incluir información necesaria e en algúin caso comprimíronse dúas entidades en unha soa como por exemplo en **Call**⁹ e **Member**¹⁰ que foron unidas dentro de **Person** como se pode ver na Figura 7.16

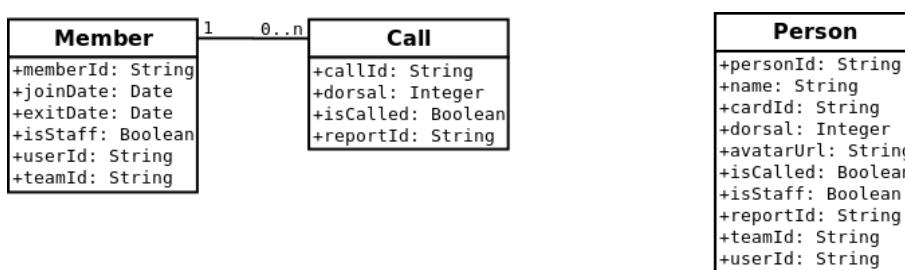


Figura 7.16: Esquema de base de datos de Person.

⁹Convocatoria dun xogador para un partido en concreto

¹⁰Membro dun equipo

7.7. I18n

Para a xestión da internacionalización da aplicación utilizouse a librería React Intl que provee dunha serie de compoñentes de React e unha API sinxela para formatear datas, números e strings, incluindo pluralización e xestión de traducións.

Esta librería dá a posibilidade de engadir un contexto aos textos que se internacionalizan, o cal facilita a labor do tradutor que se encargue de engadir ou modificar un idioma, proporcionándolle unha información extra sobre o texto que debe traducir.

Únicamente é preciso definir os compoñentes cos seus correspondentes atributos como se mostra no Fragmento de código 7.6 e a propia ferramenta xenera ficheiros *.json* seguindo o esquema de carpetas e ficheiros da aplicación con toda a información recopilada.

Listing 7.6: Exemplo de internacionalización na label dun botón.

```
<RaisedButton label={  
  <FormattedMessage  
    id='report.show.events'  
    description='Events button'  
    defaultMessage='Events' />  
} secondary={true} style={style.button}/>
```

Do mesmo xeito, a librería permite facer internacionalización de xeito programático sen utilizar compoñentes de React, a través dunha serie de funcións de Javascript.

Por último, é preciso realizar as traduccións aos diferentes idiomas, cada unha en un ficheiro de Javascript diferente, seguindo un formateado de pares cadea/valor.

7.8. Inxección de dependencias

Durante o realization do proxecto xurdiu un problema entre os servizos xa que moitos precisan dos outros para realizar comprobacións de forma circular.

A existencia destas dependencias circulares leva a que sexa imposible de determinar a relación entre eles polo que é preciso deseñar un pequeno sistema de inxección de dependencias que permite solventar esta problemática.

A solución escollida finalmente consiste en crear unha factoría seguindo o patrón Singleton [GHJV94] que se encarga de crear un obxecto para cada servizo e a continuación inxéctanse uns dentro dos outros.

Cada vez que se chama a factoría para recuperar un servizo, esta devolve o servizo coas súas dependencias inyectadas no seu interior polo que resolvemos o problema de xeito sinxelo.

Na Figura 7.7 pódese ver parcialmente o código que implementa a factoría que se encarga de crear os servizos e devovelos cando é preciso.

Listing 7.7: Fragmento da ServiceFactory.

```
let ServiceFactory = {
    isInitialized: false,
    _servicesList: new Map(),

    constructor () {
        this._personService = new PersonService()
        this._reportService = new ReportService()
        this._refereeService = new RefereeService()
        // Without dependencies
        this._teamService = new TeamService()
        // Inject dependencies
        this._eventService.ReportService = this._reportService
        this._eventService.PersonService = this._personService
        this._eventService.TeamService = this._teamService
        ...
        // Add services to the exposed list
        this._servicesList.set('ReportService', this._reportService)
        this._servicesList.set('PersonService', this._personService)
        this._servicesList.set('TeamService', this._teamService)
        this.isInitialized = true
    },

    getService (type) {
        if (!this.isInitialized) {
            this.constructor()
        }
        let service = this._servicesList.get(type)
        if (!service) {
            console.log('Error getting ', type, ' service')
        }
        return service
    }
}
```

Cómpre tamén comentar as dificultades que implica a inxección de dependencias ao realizar testing unitario xa que non se poden aproveitar as funcionalidades que aporta Jest, o framework de testing, que habitualmente permite automatizar a creación de mocks e, en este caso, obríganos a crear manualmente os mocks dos servizos e inxectalos dentro da clase a testear.

7.9. Testing e Integración continua

O testing é unha parte fundamental nun proxecto e concretamente nun proxecto de software libre no que a comunidade pode colaborar e onde poden participar persoas sen un coñecemento moi profundo da aplicación, polo que se fai totalmente imprescindible garantir que calquera cambio non rompa a integridade da mesma.

7.9.1. TDD e BDD

Neste proxecto seguironse diversas prácticas de Test Driven Development (TDD) [Mar08] e Behaviour Driven Development (BDD) [WH12] comenzando por unha definición das diversas tarefas do proxecto como tests de aceptación das funcionalidades e rematando pola realización de tests unitarios.

7.9.1.0.1. Tests de aceptación Estos tests son destinados a determinar se foron cumplidos os requisitos dunha certa funcionalidade, sempre centrándose na parte funcional e alonxándose dos detalles de implementación.

Para isto utilízanse unha linguaxe simple de dominio co fin de definir os requisitos funcionais, en este proxecto baseámonos no seguinte:

Como ... afectado ou realizador da funcionalidade.

Quero ... acción a realizar.

Cando ... momento ou caso no que debe realizarse.

Un exemplo de test de aceptación:

Como árbitro quero que o resultado se actualice automáticamente ao engadir un novo gol.

7.9.1.0.2. Tests unitarios Estos tests céntranse en comprobar o funcionamento de xeito illado dos diversos sistemas e facendo fincapé en que cada proba sexa un caso totalmente independente do resto.

Neste proxecto crearonse tests unitarios centrados nos servizos da aplicación, que son os elementos más importantes da mesma xa que son os que definen a lóxica de negocio.

Para a realización deste tipo de tests é fundamental a utilización de mocks, obxectos que imitan o comportamento de obxectos reais de xeito controlado, permitindo simular o comportamento dos obxectos dependentes.

Isto é fundamental para asegurar o illamento da funcionalidade e eliminar a dependencia de outros elementos a hora de realizar as probas unitarias.

7.9.2. Jest

É unha librería para realización de testing automático en aplicacóns ReactJS que facilita automatizar a creación de mocks ou a execución de tests cunha implementación falsa do DOM¹¹.

No Fragmento de código 7.13 pódese observar un test de exemplo do módulo de eventos e os diversos compoñentes que o forman.

¹¹Modelo en Obxectos para a Representación de Documentos

Inicialmente defínese dentro da sección `describe` unha "historia de usuario" que contén a información xeral sobre o caso de uso, dentro da cal pódense realizar diversos tests, no exemplo que vemos, o caso de uso é "Crear un evento deportivo".

Listing 7.8: Definición dunha historia de usuario en Jest.

```
describe('create Sport Event', function () {  
    ...  
}
```

Dentro do caso de uso vemos un bloque `beforeEach` que permite inicializar variables para cada novo test que se realice, eliminando de este xeito a dependencia entre os tests que as utilizan.

Dentro do bloque `describe` defínense os diversos tests desta historia de usuario, concretamente dentro dos bloques `it`.

Listing 7.9: Definición dun test en Jest.

```
it('Create a new Sport Event with valid parameters', function () {  
    ...  
}
```

Como comentábamos anteriormente, ao realizar tests unitarios é habitual a utilización de obxectos mock que imitan a funcionalidade de outros. En Jest tódolos obxectos son mocks e non é preciso definilos todos así. Cambiando un pouco a forma de traballar habitualmente, aquí débese definir ao comezo do ficheiro aqueles que non van a ser mockeados a través da sentencia definida no Fragmento de Código 7.10.

Listing 7.10: Sentencia para indicar que non se cree un mock.

```
jest.dontMock '../../../../../src/app/models/report/status/StartedStatus')
```

Tamén é habitual querer darlle un comportamento por defecto aos mocks que se utilizan, por exemplo no Fragmento de Código 7.11 estamos facendo que cando se chame a función `findById` do obxecto `reportService` co un só parámetro, o obxecto devolverá un callback cos parámetros indicados, neste caso unha Acta por defecto (`defaultReport`) e un valor nulo como segundo parámetro.

Listing 7.11: Sentencia para asignar comportamento a un mock.

```
spyOn(reportService, 'findById').andCallFake(function (anyReportId, callback) {  
    callback(defaultReport, null)  
})
```

Unha vez definido o comportamento que queremos que teñan os mocks, debemos chamar ao método real do obxecto que estamos testeando e así teremos totalmente illado o comportamento de dito método.

Finalmente utilizamos a cláusula `expect` para indicar resultados esperados, no caso de exemplo que mostramos a continuación, estamos a indicar que a chamada ao método `findById` de `reportService` foi realizada. Tamén é posible indicar valores que debe devolver, negacións ou outras validacións incluso más complexas.

Listing 7.12: Sentencia para comprobar a execución dun test.

```
expect(reportService.findById).toHaveBeenCalled()
```

A continuación mostramos o caso de test anterior completo a modo de exemplo.

Listing 7.13: Exemplo de test no módulo de eventos utilizando Jest.

```
describe('create Sport Event', function () {
  beforeEach(function () {
    defaultPerson = new Person(null, '', '', '', '', false, false, '', '', '')
    defaultTeam = new Team(null, 'Team name')
    defaultReport = new Report(null, '', '', ReportStatus.READY,
      defaultTeam, defaultTeam, [])
    defaultEvent = new EventElements.Event('event', '1', defaultPerson,
      defaultTeam, 'goal', 1, 'cause', 1)
    reportService = new ReportService(jasmine.createSpy('PersonService'),
      jasmine.createSpy('TeamService'), jasmine.createSpy('EventService'),
      jasmine.createSpy('SignService'))
    personService = new PersonService(jasmine.createSpy('ReportService'),
      jasmine.createSpy('TeamService'), jasmine.createSpy('AuthService'))
    teamService = new TeamService()
    eventService = new EventService(reportService, personService,
      teamService)
  })

  it('Create a new Sport Event with valid parameters', function () {
    spyOn(reportService, 'findById').andCallFake(function(anyReportId,callback) {
      callback(defaultReport, null)
    })

    spyOn(personService, 'findByPersonIdReportIdAndTeamId').andCallFake(function
      (anyPersonId, anyReportId, anyTeamId, callback) {
        callback(defaultPerson, null)
    })

    spyOn(teamService, 'findById').andCallFake(function (anyTeamId, callback) {
      callback(defaultTeam, null)
    })

    spyOn(EventDao, 'create').andCallFake(function (reportId, person, team,
      eventType, matchTime, cause, timestamp, callback) {
      callback(defaultEvent, null)
    })

    eventService.create(defaultEvent.reportId, defaultEvent.person,
      defaultEvent.team, defaultEvent.type, defaultEvent.matchTime,
      defaultEvent.text, defaultEvent.timestamp,
      (event, err) => {
        expect(event).toEqual(defaultEvent)
        expect(event).not.toBe(null)
        expect(err).toBe(null)
      })
  })
})
```

```
expect(reportService.findById).toHaveBeenCalled()
expect(personService.findByPersonIdReportIdAndTeamId).toHaveBeenCalled()
expect(teamService.findById).toHaveBeenCalled()
expect(EventDao.create).toHaveBeenCalled()
}))})
```

7.9.3. Travis CI

A medida que a aplicación foi medrando, xurdíu a necesidade de simplificar traballos como a búsquedas de erros ou o mantemento da aplicación, buscando garantir que se mantén a integridade da mesma en todo momento e que non se introducen erros polo que se decidíu engadir tests unitarios na aplicación.

Da mesma maneira lanzouse unha primeira versión estable do proxecto polo que tamén se decidíu engadir un sistema de integración continua ao mesmo co fin de garantizar que a aplicación xerada na rama de producción é funcional, compila en todo momento e pasa tódolos tests.

Para isto utilizóuse Travis CI como sistema integración para o que se engadíu un ficheiro “*.travis.yml*” no que se lle indican diversos parámetros de configuración, como as versións de Javascript coas que debe poñer a proba o funcionamento do programa, os scripts a executar para instalar dependencias e para executar o programa, as ramas de desenvolvemento sobre as que debe actuar, no noso caso `master` e `development` ou as notificacións que debe enviar ao finalizar un traballo, por exemplo a través de correo electrónico ou utilizando un servizo de mensaxería instantánea como Slack.

Nesta integración xurdíu un problema derivado do sistema de construción utilizado no proxecto, Gulp, que nos axuda a automatizar tarefas como a compilación, a execución dos tests ou o deploy da aplicación, e derivado tamén do framework de tests, Jest.

Cando este último se executaba a través de Gulp, non devolvía os erros ao resolver os tests como resultado dunha función, se non que o facía a través dun callback. É por iso polo que foi preciso engadir o control de este caso para evitar falsos positivos na execución dos tests unitarios dende Travis, dentro da tarefa “test” definida en Gulp.

Capítulo 8

Conclusións e traballo futuro

Índice xeral

8.1. Recoñecementos	79
8.1.1. Finalista no Certamen de Proyectos Libres da UGR	79
8.1.2. Premio Universitario de Software Libre	80
8.2. Traballo futuro	80
8.2.1. Melloras de desenvolvemento	80
8.2.2. Creación de comunidade	81
8.3. Conclusións	82

NESTE capítulo contaremos as conclusións obtidas da realización do proxecto, os recoñecementos obtidos polo mesmo e as liñas de traballo futuro que temos plantexadas.

8.1. Recoñecementos

Durante a realización do proxecto tamén se participó en varios certames e concursos, non só coa idea de obter recoñecementos se non tamén co obxectivo de difundir o proxecto e buscar colaboradores para a comunidade de software libre que estamos a crear.

8.1.1. Finalista no Certamen de Proyectos Libres da UGR

O día 6 de Xuño celebrouse na Facultade de Informática da Universidade de Granada (UGR) a final do “Certamen de Proyectos Libres” que organiza a Oficina de Software Libre de dita universidade dende xa fai varios anos.

VACmatch Mobile convertíuse en un dos 7 finalistas e foi invitado a participar presentando o proxecto ante un xurado composto por diversos profesionais do sector, presentación que se realizou en video ao non poder asistir ao evento personalmente.

8.1.2. Premio Universitario de Software Libre

Participouse no Concurso Universitario de Software Libre, un certame no que competiron ata 75 persoas con máis de 40 proxectos de software libre de todo España e no que VACmatch Mobile foi premiado.

Durante o mes de Maio celebróuse na Facultade de Informática da Universidade de Sevilla a fase final nacional na que fomos invitados xunto cos outros tres finalistas, a participar e expoñer o noso proxecto ante diversos profesionais e empresas do sector.

Finalmente recibimos o premio ao Mellor Proxecto para Dispositivos Móbiles con unha remuneración de 300 € en metálico e por suposto todos os gastos do desplazamento e estancia durante a fase final foron cubertos pola organización do concurso.

8.2. Traballo futuro

Como calquera aplicación actual, este proxecto non é un proxecto totalmente concluído xa que certos requisitos iniciais non foron aínda implementados e múltiples deportes poden ser engadidos.

Tamén gracias a utilización de metodoloxías áxiles de desenvolvemento, ao longo do proxecto surdiron novas propostas, das cales algunas non foron aínda implementadas e forman parte de liñas de traballo futuro.

8.2.1. Melloras de desenvolvemento

8.2.1.0.1. Creación dunha versión de demostración Sería interesante engadir unha funcionalidade para realizar entregas continuas que permita automatizar a posta en producción dunha versión do programa como demostración para posibles federacións interesadas en coñecer o funcionamento da aplicación.

8.2.1.0.2. Resolución de conflictos Actualmente cando un elemento se modifica ao mesmo tempo en dous lugares a vez (un acta na federación e no campo, por exemplo), os conflictos non son resoltos se non que se almacenan ambas versións para escoller cal sería a correcta, polo que sería interesante proveer aos usuarios dunha interfaz para resolver ditos conflictos de xeito sinxelo.

8.2.1.0.3. Integración con VACmatch Web O punto forte da aplicación sería poder integrar o sistema de xestión de competicións de VACmatch coa aplicación móvil co fin de simplificar aínda en maior medida o traballo do árbitro e do xestor da federación deportiva.

8.2.1.0.4. Módulo para a xestión multifederación Outra funcionalidade interesante sería a de engadir a posibilidade de realizar a autenticación dos árbitros contra diversos hosts de federacións, pensando na opción de que cada federación poida ter unha instancia propia da base de datos.

8.2.1.0.5. Módulo de notificacións Sería interesante que o árbitro recibise notificacións sobre cando a aplicación sincroniza novas actas de encontros, co fin de facilitarlle coñecer o lugar, data e hora a onde se debe desplazar para árbitrar e incluso engadir un sistema de recordatorios dos encontros que ten asignados.

8.2.2. Creación de comunidade

Nun proxecto de software libre é fundamental dispor dunha comunidade de desenvolvedores que axuden a mantelo vivo polo que tamén queremos mencionar diversas liñas de traballo a seguir en este punto.

8.2.2.0.1. Hackathons de desenvolvemento. Organizaranse diversos hackathons de desenvolvemento de 1 ou 2 días de duración nos que presentar o proxecto e tratar de buscar desenvolvedores para impulsar unha pequena funcionalidade ou un pequeno prototipo ao redor de VACmatch co fin de introducilos no proxecto.

8.2.2.0.2. Proxectos de Fin de Grao ou Master. Traballarase con profesores e asociacións para promover proxectos de fin de grao e master baseados en VACmatch, en lugar de crear pequenas aplicacións a medida para unha asociación ou federación, traballar sobre un proxecto grande e múltideporte.

8.2.2.0.3. GPUL Summer of Code. A asociación GPUL está desenvolvendo un programa de apoio a proxectos de software libre para que estudiantes universitarios colaboren en ditos proxectos durante un verán polo que trataremos de propor VACmatch como un dos proxectos no que estos estudiantes poidan colaborar.

8.3. Conclusíons

O software libre é fundamental na sociedade actual, a inmensa maioría dos avances tecnolóxicos baseanse en solucións libres e cada vez son máis os que as desenvolven.

No campo da tecnoloxía aplicada ao deporte, pouco a pouco comézanse a ver as primeiras solucións para informatizar a xestión pero, concretamente o software libre, atópase aínda con un longo camiño por recorrer e con este proxecto conseguimos sentar unhas bases para isto, creando a primeira aplicación libre para xestionar actas electrónicas de encontros deportivos.

VACmatch Mobile permite que os árbitros das competicións deportivas poidan xestionar as actas directamente no seu teléfono móvil e estas sincronizar en tempo real coa páxina web da federación, mantendo os resultados actualizados en todo momento e permitindo incluso cubrir as actas cando non dispoñen de conexión a internet.

A aplicación é multiplataforma e foi testeada por usuarios reais polo que está validada por profesionais para ser utilizada nun contexto real, así mesmo atópase escrita pensando na extensibilidade, facilitando que os desenvolvedores interesados polo proxecto, poidan ampliala de xeito sinxelo e engadir novas funcionalidades, sobre todo, deportes.

Apéndices

Apéndice A

Configuración e execución da aplicación.

VACmatch Mobile é Software Libre e o seu código fonte atópase publicado no seguinte repositorio de código en GitHub github.com/vacmatch/vacmatch-mobile.

Para poder executar a aplicación é preciso dispor das seguintes ferramentas:

- Git
- Gulp
- NodeJS
- NPM
- CouchDB

A.1. Execución dende plataforma Web.

O primeiro paso é abrir o ficheiro de configuración da aplicación situado en `src/app/api/config.json` e nel atoparemos a configuración das bases de datos remotas para os entornos de desenvolvemento e producción.

É preciso dispor dunha base de datos CouchDB funcionando ben en local ou ben en remoto e no ficheiro de configuración indicado anteriormente podemos introducir os datos de conexión e os nomes das coleccións que se van a utilizar para a autenticación e para o resto da aplicación.

Modificando a variable `_env` podemos cambiar de entorno e a través de `_type` podemos seleccionar se a compilación se vai a realizar para unha aplicación web ou móvil.

Unha vez arrancada a aplicación, únicamente é preciso abrir un navegador web e acceder a `localhost:8080/#/login`

Listing A.1: Configuración e execución web

```
// Instalar dependencias
> npm install
// Compilación
> gulp build
// Arrancar servidor
> gulp run
```

A.2. Execución dende plataforma móvil.

Unha vez configurado proxecto, o comando `gulp deploy` constrúe e copia a aplicación dentro da carpeta de Apache Cordova.

Alí débese engadir a plataforma que se deseñe utilizar, constuir a aplicación para dita plataforma e arrancala de xeito sinxelo.

Listing A.2: Configuración e execución móvil

```
// Instalar dependencias
> npm install

// Compilación
> gulp deploy

// Entrar na carpeta de cordova, engadir plataforma android e construir a
// aplicación
> cd cordova-app

> cordova platform add android

> cordova build android

// Arrancar en móvil
> cordova run android

// Arrancar nun emulador
> cordova emulate android
```

Apéndice B

Glosario de acrónimos

IEBT *Iniciativa Empresarial de Base Tecnolóxica.*

FGVB *Federación Galega de Voleibol.*

API *Application Programming Interface.*

REST *Representational State Transfer.*

TDD *Test Driven Development.*

BDD *Behaviour Driven Development.*

DOM *Document Object Model.*

HTML *HyperText Markup Language.*

CSS3 *Cascading Style Sheets.*

CUSL *Concurso Universitario de Software Libre.*

SPA *Single Page Application.*

Apéndice C

Glosario de términos

VACmatch Web VACmatch Web é unha plataforma de xestión de competicións deportivas que permite realizar todo tipo de trámites coas federacións deportivas de forma electrónica e reduce enormemente o traballo que estas deben realizar no seu día a día.

VACmatch Mobile É unha aplicación que permite que os árbitros deportivos poidan xestionar as actas dos seus encontros de forma electrónica.

Actas É o lugar onde se almacena a información sobre un encontro deportivo, inclue os equipos, os lugares onde se xogou e o resto de estadísticas de cada xogador durante o partido.

Fichas Unha ficha é un documento con fotografía incluida que identifica a un xogador que compite nunha competición e que debe levar a tódolos encontros para poder disputar os partidos.

Xestor da competición Persoa encargada da xestión do calendario, da recepción das actas dos encontros, da súa revisión, da súa publicación e, en xeral, da xestión dunha competición.

Árbitro Persoa que se encarga de velar polo cumplimento do regulamento dun deporte durante un encontro e así mesmo debe tomar nota na acta, das estadísticas e dos diversos eventos que ocurren nun encontro.

API Interfaz de programación de aplicacións que abstrae unha serie de funcións para a súa utilización dende outro software.

Rest É un tipo de arquitectura de desenvolvemento web que se basea no protocolo HTTP coa idea de utilizar os diversos verbos que define o protocolo para interactuar cos recursos de unha API.

Lean Startup Metodoloxía de desenvolvemento de negocio centrada no cliente e na aprendizaxe validada.

eXtreme Programming Metodoloxía de desenvolvemento de software que se centra na necesidade de adaptarse aos cambios no avance do proxecto propoñendo prácticas como testear antes de programar, revisión de código, programación en parellas ou simplicidade no desenvolvemento entre outras.

Scrum Metodoloxía de desenvolvemento de software áxil centrada en mellorar a xestión dun proxecto a través dunha serie de prácticas de revisión e xestión por iteracións de desenvolvemento chamadas sprints.

Sprint Iteración de desenvolvemento dentro das metodoloxías áxiles que habitualmente ten unha duración de entre 1 e 3 semanas.

Release Versión entregable do produto.

Pull Request Petición de integración dunha rama de Git dentro doutra. É a forma habitual de colaborar en proxectos que se atopan en GitHub.

Startup É un tipo de empresa en construción, habitualmente con produto propio, moi áxil e que busca operar con uns costos mínimos e obter unhas ganancias de medren exponencialmente.

Javascript Linguaxe de programación interpretada, moi habitual para desenvolver aplicacóns web no lado do cliente, aínda que últimamente está a ganar protagonismo tamén no servidor.

SQLite Sistema de xestión de bases de datos relacional, moi lixeiro e utilizado habitualmente en aplicacións móbilas nativas.

Mock É un obxecto que simula o comportamento de outro obxecto real, útil para a realización de tests unitarios.

Bibliografía

- [Bod16] Adam Boduch. *Flux Architecture*. Packt Publishing, May 2016.
- [DGG12] Rafael de las Heras del Dedo, Carmen Lasa Gómez, and Alonso [et al] Álvarez García. *Métodos Ágiles y Scrum*. Anaya Multimedia-Anaya Interactiva, Madrid, January 2012.
- [Eis] Bonnie Eisenman. *Learning React Native: Building Native Mobile Apps with JavaScript*. O'Reilly Media, 1 edition edition.
- [GHJV94] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison Wesley, Reading, Mass, 01 edition edition, October 1994.
- [Mar08] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, Upper Saddle River, NJ, 1 edition edition, August 2008.
- [MP] Michael Mikowski and Josh Powell. *Single Page Web Applications: JavaScript end-to-end*. Manning Publications, edición: 1 edition.
- [Pan16] Mahesh Panhale. *Beginning Hybrid Mobile Application Development*. Apress, 1st ed. 2016 edition edition, May 2016.
- [Rie11] Eric Ries. *The Lean Startup: How Constant Innovation Creates Radically Successful Businesses*. Portfolio Penguin, London; New York, October 2011.
- [SG15] Andrew Stellman and Jennifer Greene. *Learning Agile: [understanding Scrum, XP, Lean, and Kanban]*. O'Reilly, Beijing, 1st ed edition, 2015. OCLC: 897773274.
- [Tiw11] Shashank Tiwari. *Professional NoSQL*. Wrox, Indianapolis, IN, 1 edition edition, September 2011.
- [WH12] Matt Wynne and Aslak Hellesoy. *The Cucumber Book: Behaviour-Driven Development for Testers and Developers*. Pragmatic Bookshelf, Dallas, Tex, 1 edition edition, February 2012.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but
changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, TeXinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is preci-

sely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent

steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
 - B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
 - C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
 - D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
-

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as

Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special

permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDEDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.