

UNIVERSIDADE DE SANTIAGO DE
COMPOSTELA



ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Título do Traballo de Fin de Grao

Subtítulo do Traballo de Fin de Grao

Autor:

Pablo Pérez Román

Directores:

Paulo Félix Lamas

David González Márquez

Grao en Enxeñaría Informática

Setembro 2014

Traballo de Fin de Grao presentado na Escola Técnica Superior de Enxeñaría
da Universidade de Santiago de Compostela para a obtención do Grao en

Enxeñaría Informática



D. Paulo Félix Lamas, Profesor do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela, e **D. David González Márquez**, Profesor do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela,

INFORMAN:

Que a presente memoria, titulada *JDataMotion: unha ferramenta para a visualización dinámica de diagramas de dispersión*, presentada por **D. Pablo Pérez Romaní** para superar os créditos correspondentes ao Traballo de Fin de Grao da titulación de Grao en Enxeñaría Informática, realizouse baixo nosa dirección no Departamento de Electrónica e Computación da Universidade de Santiago de Compostela.

E para que así conste aos efectos oportunos, expiden o presente informe en Santiago de Compostela, a 08/09/2014:

O director,

O codirector,

O alumno,

Paulo Félix Lamas David González Márquez Pablo Pérez Romaní

Agradecimentos

Se se quiere pór algún agradecemento, este vai aquí.

Resumo

Se se quiere pór resumo, este vai aquí.

Índice xeral

1. Introducción	1
1.1. Obxectivos xerais	2
1.2. Relación da documentación	2
1.3. Descrición do sistema	3
2. Xestión do proxecto	5
2.1. Xestión de riscos	5
2.2. Metodoloxía de desenvolvemento	9
2.3. Planificación temporal	11
2.4. Xestión da configuración	14
2.5. Análise de custos	15
3. Análise de requisitos	17
3.1. Casos de uso	17
3.2. Requisitos funcionais	18
3.3. Requisitos de calidade	29
3.4. Requisitos de deseño	29
3.5. Requisitos non funcionais	30
3.6. RFs dos sprints	31
4. Deseño	35
5. Exemplos	37
5.1. Un exemplo de sección	37
5.1.1. Un exemplo de subsección	37
5.1.2. Outro exemplo de subsección	37
5.2. Exemplos de figuras e cadros	38
5.3. Exemplos de referencias á bibliografía	39
5.4. Exemplos de enumeracións	39
6. Conclusións e posibles ampliacións	41
A. Manuais técnicos	43

B. Manuais de usuario	45
C. Licenza	47
Bibliografía	49

Índice de figuras

2.1. Diagrama de Gantt	12
2.2. Esquema de Descomposición do Traballo (EDT)	13
5.1. Esta é a figura de tal e cal.	38

Índice de cadros

2.1. Custos	15
5.1. Esta é a táboa de tal e cal.	38

Capítulo 1

Introdución

Na actual sociedade da información, onde a cantidade de datos que se manexan aumenta día a día de xeito exponencial, a minería de datos convértese nunha ferramenta fundamental para poder explotalos de maneira eficaz, co fin último de xerar coñecemento a partir dos mesmos.

Para visualizar estes datos unha das técnicas máis utilizadas son os diagramas de dispersión ou scatterplots. Estes permítenos analizar os datos e atopar con facilidade relacións entre as distintas variables, como a correlación entre elas, a distribución dos puntos no plano, a tendencia dos datos recollidos ou outras características que sería complicado extraer a partir dun simple listado, posiblemente desordenado, de tuplas de información. Non obstante, os scatterplots restrínxennos a unha perspectiva estática do problema. En moitos deses problemas imos encontrar unha compoñente dinámica intrínseca como é o tempo. Con este proxecto pretendemos dotar a esta representación da súa perspectiva dinámica, para amosar os datos engadindo outro punto de vista que enriqueza a información extraída.

A ferramenta pretende etiquetar cada punto dun scatterplot cun valor de significado temporal, de tal xeito que este puidese ser empregado en funcións de reprodución. Este valor numérico podería referenciar dende o momento de captación da tupla que a contén, ata unha ordenación dos datos atendendo á súa prioridade ou relevancia. A utilidade de unha ferramenta como esta xa foi validada no campo da electrocardiografía, pero sen ningunha dúbida haberá moitos outros ámbitos que poderán sacar proveito da análise de datos desde unha perspectiva dinámica.

1.1. Obxectivos xerais

A motivación principal deste proxecto é o desenvolvemento dunha ferramenta capaz de visualizar a modo de filme a evolución dun conxunto de datos ao longo dunha magnitude como sería o tempo, ademais de permitir o preprocesado ou manipulación deses datos. Sendo máis específicos, este proxecto busca a realización da análise, deseño e implementación dunha aplicación que consiga:

- Facilitarlle ao usuario o procesado de volumes de datos dun tamaño significativo.
- Posibilitar o traballo con formatos de arquivo CSV ou ARFF.
- Dispor das funcionalidades necesarias para manipular os datos.
- Ser capaz de amosar os datos en forma de scatterplots, con funcións de reprodución básicas. Tamén se debe posibilitar a configuración desta reprodución por parte do usuario.
- Aplicar filtros nos datos cos que se traballa.
- Interaccionar co usuario por medio dunha interface simple e amigable coa que se sinta identificado.

Así mesmo, outros obxectivos colaterais son:

- Aplicar nun caso real a ferramenta JDataMotion, para apreciar a súa utilidade.
- Finalizar o desenvolvemento do proxecto antes do día 8 de Setembro de 2014

1.2. Relación da documentación

Esta memoria plasma o proceso de desenvolvemento do proxecto JDataMotion, que persegue os obxectivos citados no apartado anterior.

Os distintos capítulos repártense do modo que segue:

Capítulo 1. Introducción: composta por obxectivos xerais, relación da documentación que conforma a memoria, descrición do sistema (métodos, técnicas ou arquitecturas utilizadas e xustificación da súa elección).

Capítulo 2. Planificación e presupostos: inclúe a estimación do custo (presuposto) e dos recursos necesarios para desenvolver este proxecto, xunto coa planificación temporal do mesmo e a división en fases e tarefas.

Capítulo 3. Especificación de requisitos: inclúe a especificación do Sistema, xunto coa información que este debe almacenar e as interfaces con outros Sistemas, sexan hardware ou software, e outros requisitos (rendemento, seguridade, etc).

Capítulo 4. Deseño: rexistra como se realiza o Sistema, a división deste en diferentes compoñentes e a comunicación entre eles. Así mesmo, neste apartado determínase o equipamento hardware e software necesario.

Capítulo 5. Exemplos.

Capítulo 6. Conclusións e posibles ampliacións.

Apéndice A. Manuais técnicos: incluírase toda a información precisa para aquelas persoas que se vaian a encargar do desenvolvemento e/ou modificación do Sistema.

Apéndice B. Manuais de usuario: incluírán toda a información precisa para aquelas persoas que utilicen o Sistema: instalación, utilización, configuración, mensaxes de erro, etc.

Apéndice C. Licenza.

Bibliografía

1.3. Descrición do sistema

A ferramenta JDataMotion desenvolverase integramente na linguaxe de programación Java, posto que necesitamos unha linguaxe orientada a obxectos que axilice o desenvolvemento do software, favoreza a reutilización de código e facilite o deseño dunha interface gráfica. Dentro do paradigma orientado a obxectos, Java é unha solución razoable que ademais conferiría á nosa aplicación bastantes opcións á hora de representar os scatterplots, grazas ás librerías de terceiros que fornecen esta funcionalidade. Por outra banda, podemos usar a librería gráfica Swing para a implementación da interface.

A nivel funcional, JDataMotion busca estender coa perspectiva dinámica as posibilidades do software de Weka[1]. En base a isto, intentaremos adaptar algunhas das súas funcionalidades e incluso botaremos man da súa interface de programación (API), sobre todo na parte do modelo da aplicación.

O seu funcionamento parte dun arquivo dado en formato CSV ou ARFF que se deberá importar nun primeiro momento, ou ben dunha sesión (en formato JDM) gardada durante un experimento anterior. Botarase man das librerías facilitadas pola ferramenta Weka[1] para a importación, exportación e almacenamento

do modelo de datos.

Os tipos de datos dos atributos poderán ser configurados de acordo ás seguintes etiquetas propias do estándar que proporciona o formato ARFF[3]: nominal, numérico, string ou data. A maiores, tamén se poderá especificar que un atributo numérico actúe como índice temporal para ser utilizado na reprodución. Así mesmo, permítese a inserción, eliminación e modificación dos datos.

Poderanse engadir filtros configurables aos datos que se están a procesar. Os filtros representaranse nunha secuencia, e permitirase a adición, desprazamento ou eliminación de filtros nela. Tamén se facilitará unha interface pública para que calquera desenvolvedor poida aplicar no seu experimento filtros personalizados, sempre que implementen esa interface.

De acordo coa ferramenta Weka[1], para a visualización de datos seguirase un esquema matricial, de xeito que para cada par de atributos numéricos, exista unha cela dentro de esa matriz para representalos baixo a forma dun scatterplot, de tal forma que os scatterplots dentro da matriz estarán ordenados por filas e columnas segundo o atributo que representen en cada eixo. Para a creación de scatterplots escolleise JFreeChart[2], debido ás súas prestacións[2][4]:

- É unha solución desenvolvida en código aberto, e distribuído baixo licencia pública LGPL.
- A interface de programación (API) está extensamente documentada, o cal facilita a aprendizaxe do seu uso.
- Da soporte moitos tipos de gráficas, non só scatterplots, o cal será útil á hora de amosar histogramas para resumir variables.
- As gráficas xa implementan funcións de zoom e reposicionamento (automáticos e manuais), así como unha gran serie de opcións de personalización (cores, liñas, puntos, etc.) ou incluso a posibilidade de exportar a nosa imaxe en formato PNG ou JPEG.
- Permite a creación de gráficas dinámicas, isto é, permite engadir en tempo de execución puntos ás gráficas, o cal constituirá o punto de partida para desenvolver as funcións de reprodución.

Ademáis, a reprodución dinámica dos datos deberá ser configurable. Poderase sinalar un atributo nominal para que os seus valores se representen con puntos de cor e forma diferente e ampliar un scatterplot nunha nova ventá, así como facer zoom e reposicionar a ventá de cada scatterplot.

O usuario poderá exportar o seu traballo en calquera momento baixo un novo ficheiro de formato ARFF ou CSV, ou ben gardar a sesión (JDM) para retomala máis adiante.

Capítulo 2

Xestión do proxecto

Neste capítulo comentaremos distintos aspectos relacionados coa planificación de como se vai xestionar este proxecto. Falaremos, por exemplo, da xestión de riscos que conleva o desenvolvemento do software, así coma os métodos de continxencia, prevención ou minimización que seguiremos en caso da incidencia dos mesmos. Cos riscos expostos, abordaremos a metodoloxía de desenvolvemento máis axeitada para o proxecto, de acordo tamén cos obxectivos anteriormente plasmados. Seguiremos coa planificación temporal do proxecto e finalizaremos coa estimación de custo e prazos, así como a xestión da configuración.

2.1. Xestión de riscos

Na fase de planificación dun proxecto hai que sopesar os distintos riscos aos que estará exposto o seu desenvolvemento, cuantificalos e deseñar estratexias para a súa aparición. Algúns dos riscos nun Traballo de Fin de Grao poden ter graves consecuencias na liña base do proxecto debido á inexperiencia do seu autor, polo que fronte á falta de experiencia hai que esforzarse en mellorar a planificación.

No análise de riscos valoraremos a probabilidade de aparición e a súa gravidade, para a continuación deseñar unha medida de continxencia, prevención ou minimización. A escala de valoración da probabilidade e da gravidade vai ser:

- Moi baixa
- Baixa
- Media
- Alta
- Moi alta

Os riscos considerados son os seguintes:

■ **Risco 01**

Nome: Cambios no alcance durante o desenvolvemento

Descrición: A lista inicial de requisitos funcionais que se captará nas primeiras reunións cos titores vai sufrir modificacións, incluso co proxecto en etapas avanzadas de desenvolvemento. A súa probabilidade duplícase pola dobre titularidade do Traballo de Fin de Grao.

Probabilidade: Moi alta

Gravidade: Alta

Medidas de minimización: Botaremos man da folgura temporal do proxecto (marxe de tempo dispoñible para eventualidades). Os cambios razoaranse cos titores, presentando a lista de requisitos actual e valorando a parte da folgura que consumirían ditos cambios. Tamén se tratará de ter reunións de avaliación cos titores cunha alta frecuencia, para así detectar o antes posible calquera cambio nos requisitos, se ben pode acontecer que se propoñan cambios sobre as primeiras etapas cando o proxecto se atopa en etapas avanzadas.

■ **Risco 02**

Nome: Confrontación de opinións entre os distintos titores

Descrición: Un cambio ou opinión dun titor contradí á do outro.

Probabilidade: Moi alta

Gravidade: Media

Medidas de prevención: Trataremos de citarnos cos dous titores á vez, de xeito que calquera dualidade de opinións se resolva de xeito presencial. Nestes casos o risco non será considerado como acontecido.

Medidas de minimización: Cando as reunións non se podan realizar cos dous titores á vez, informarase ao outro o antes posible das decisións, acordos ou cambios prantexados, así poderemos coñecer a súa opinión antes de obrar en consecuencia. Intentarase que toda esta comunicación externa ás reunións se realice por correo electrónico, e tratarase de incluír a todos os participantes (menos ao propio emisor) no grupo de destinatarios das mensaxes. Ademais, deste xeito quedará rexistrada a resposta e evitase o repudio.

■ **Risco 03**

Nome: Imprecisión á hora de fixar entregables

Descrición: A inexperiencia do alumno manifestarase xa nas primeiras entregas programadas. Ao non ter traballado previamente en proxectos desta índole, resultará complicado estimar os prazos de entrega nas primeiras fases do proxecto, tanto por exceso como por defecto.

Probabilidade: Alta

Gravidade: Media

Medidas de minimización: Intentaremos especificar entregas dun contido menor e máis frecuentes, sobre todo nas primeiras fases, para que sexa máis doado comezar a estimar correctamente os prazos de entrega.

■ **Risco 04**

Nome: Imposibilidade de reunirse cun dos titores

Descrición: Un dos titores non pode acudir a algunha reunión proposta, nin estará nos seguintes 5 días.

Probabilidade: Alta

Gravidade: Baixa

Medidas de minimización: Desenvolverase a reunión co titor dispoñible, sendo mester, de acordo co Risco 02, informar das conclusións sacadas ao outro titor por medio do correo electrónico en canto remate a reunión.

■ **Risco 05**

Nome: Imposibilidade de reunirse con ningún dos dous titores

Descrición: Ningún dos titores está dispoñible para unha reunión, nin estará nos seguintes 5 días.

Probabilidade: Media

Gravidade: Media

Medidas de minimización: Tratarase de programar unha videoconferencia con polo menos un dos dous titores. No caso dun só titor, abordaranse as medidas adoptadas para o Risco 4. Se isto non fose posible, continuaríase traballando no seguinte entregable ata que algún dos titores volvese estar dispoñible.

■ **Risco 06**

Nome: Descoñecemento ou inexperiencia coas solucións

Descrición: O alumno non coñece as posibilidades que teñen as ferramentas das que dispón (librarías, módulos, solucións, etc.).

Probabilidade: Alta

Gravidade: Media

Medidas de prevención: Adicarase un tempo prudencial, nas primeiras fases, a revisar as APIs e a documentación en xeral das librerías, proxectos de terceiros e demais ferramentas que se van empregar, para ser conscientes de como poden solucionar as nosas necesidades.

■ Risco 07

Nome: Mala escalabilidade do sistema

Descrición: O noso sistema escala mal con entradas de datos de gran tamaño.

Probabilidade: Alta

Gravidade: Media

Medidas de minimización: Programaremos prazos de tempo maiores para entregas nas que a escalabilidade poida ser un problema (visualización, carga de datos, etc.) para mellorar este aspecto. Se non é viable reducir máis a latencia, procederase a aliviala mellorando a usabilidade do sistema (por exemplo, empregando barras de progreso).

■ Risco 08

Nome: Imposibilidade de finalizar o proxecto en tempo

Descrición: A folgura está esgotada, e o cumprimento do prazo de entrega vese ameazado.

Probabilidade: Media

Gravidade: Moi alta

Medidas de prevención: Evitaremos na medida do posible recorrer á folgura, e trataremos de seguir a planificación do xeito máis estrito que podamos.

Medidas de minimización: Poremos en coñecemento aos titores do estado do proxecto e dos seus prazos, para discutir a modificación ou eliminación dalgúns ítems da especificación.

2.2. Metodoloxía de desenvolvemento

A elección da metodoloxía de traballo é un paso importante na planificación de calquera proxecto, xa que a posteriori influirá en varios aspectos deste: a xestión dos seus riscos, a súa tolerancia a cambios externos, a confianza na súa validez, etc. Hai dous enfoques fundamentais: as metodoloxías estritas e as metodoloxías áxiles. As primeiras esixen unha planificación estrita, practicamente inmutable e necesariamente realista de todo o plan de traballo, e son boas cando o conxunto de requisitos é fixo e moi concreto. As segundas, pola contra, son flexibles e adaptanse ben a cada situación, pois nelas asúmese que se van producir variabilidades nos requisitos.

Sopesando as circunstancias nas que se desenvolve un Traballo de Fin de Grao, onde a experiencia do alumno é practicamente nula no que respecta a xestión de proxectos, e tendo en consideración dous clientes (titores) aos que atender, semella que deberíamos adoptar unha metodoloxía de traballo que se adapte ás necesidades de cambios que vaian xurdindo, e que consiga en cada entrega recibir certo 'feedback' por parte dos titores, de forma que tras cada iteración podamos ter a seguridade da correspondencia entre o proxecto e o modelo mental de quen o especificou. É dicir, necesitamos unha metodoloxía áxil.

Dentro do compendio de metodoloxías áxiles existentes, decantarémonos pola metodoloxía Scrum[7], pois enfoca todas as súas avaliacións sobre entregas parciais, pero funcionais, para facilitarlle ao receptor do proxecto a valoración do mesmo. Necesítase, polo tanto, unha gran implicación do cliente no proxecto, algo que se pode conseguir dada a dualidade da tutoría (é máis doado que haxa un titor dispoñible para realizar a reunión). Por outra banda, os requisitos que constitúen as distintas entregas deben estar priorizados para que o proxecto poida avanzar cun carácter incremental, e ditas entregas deben resultar usables para o cliente.

Scrum define unha serie de ferramentas e de regras, idóneas para levar a cabo o desenvolvemento de proxectos que buscan unha metodoloxía áxil. Os preceptos básicos deste modelo son:

- Adoptar unha estratexia de desenvolvemento incremental, no canto da planificación e execución completa do produto (é dicir, no canto dunha metodoloxía estrita).
- Basear a calidade do resultado máis no coñecemento das persoas que o especificaron ca na calidade dos procesos empregados.
- Solapamento das fases de desenvolvemento (análise, deseño, implementación e probas) no canto da sucesión secuencial que nos ofrecen metodoloxías como a cascada.

A metodoloxía Scrum comeza cunha captación de requisitos en reunión co cliente, da cal se extrae un Backlog, é dicir, unha lista ordenada por prioridade de requisitos funcionais (RFs). Ademais, Scrum define o sprint como unidade elemental de tempo de traballo. Un sprint dura entre 1 e 4 semanas, aínda que nós trataremos de manter a súa duración en 2 ou incluso 1 semanas para maximizar a supervisión e xestionar ben os riscos, sobre todo nas etapas iniciais.

Ao término de cada reunión co cliente, revísase o Backlog e incorpórase un certo número de RFs a un novo sprint, o cal dará comezo en canto remate a reunión. Para o remate dese novo sprint (dentro dunha semana no noso caso) terase programada a seguinte reunión, na que se valorará o sprint finalizado e se accederá ao Backlog para acordar o seguinte sprint, e así sucesivamente. A valoración do sprint en cada reunión realízase presentando a lista de RFs de dito sprint, e demostrando ante o receptor do proxecto que cada un dos ítems ou tarefas do sprint funciona correctamente.

Para regular o desenvolvemento desa metodoloxía pódese botar man de diversas ferramentas, de entre as cales nós escollemos Acunote[8] para o noso proxecto. Acunote é unha aplicación web especialmente deseñada para a xestión da metodoloxía Scrum. Ten varios plans de prezos, pero nós empregaremos o gratuíto porque as nosas necesidades restrínxense a un equipo de persoal pequeno (o alumno e os dous titores). Entre as prestacións desta ferramenta, sacarémoslle maior proveito ás seguintes:

Wiki: Empregarémola para engadir contido visible ao resto de membros do grupo.

Lista de sprints: Amosa os sprints en 3 grupos: sprints pasados, sprints presentes e sprints futuros. Ao abrir un sprint visualízanse os ítems ou tarefas (requisitos funcionais no noso caso) que o compoñen. Accederemos a este apartado na maioría dos casos para crear novos sprints.

Sprint actual: Visualiza os RFs do sprint actual. A medida que se vaian completando requisitos funcionais, accederase a esta lapela para cambiar o estado do requisito en cuestión. Os estados posibles son:

- Non comezado (por defecto)
- En progreso
- Reaberto
- Bloqueado
- Completado
- Verificado
- Duplicado

- Non se vai realizar

Backlog: Contén todos os ítems (requisitos funcionais) pendentes de ser asignados a un sprint. Este lista cumprimentarase ao principio, cos requisitos funcionais captados e ordenados por prioridade, e logo accederase a ela á hora de asignar RFs aos novos sprints. Na extracción de RFs débese respectar a orde dos mesmos dentro da lista, collendo sempre un número de ítems determinado da parte superior. Estes ítems desaparecerán do Backlog en canto sexan asignados.

Tarefas: Mostra todos os ítems especificados, independentemente de que fosen asignados a un sprint ou non.

2.3. Planificación temporal

A metodoloxía Scrum caracterízase como ben dixemos polo solapamento das fases que nun modelo en cascada estarían ben separadas. As primeiras semanas de traballo estarán adicadas á análise para a captación inicial de requisitos, pero nas sucesivas iteracións ou sprints poderán realizarse en paralelo análise, deseño, implementación e probas. Esta é unha das licencias que outorga o emprego das metodoloxías áxiles. De todos xeitos, aínda dentro da variabilidade destas metodoloxías, podemos dividir a vida do proxecto en unha serie de fases fundamentais:

Inicio: Constitúe o primeiro sprint (Sprint 00) da planificación, e durará dúas semanas. Nesta fase programaranse reunións cos titores para realizar a captación de requisitos funcionais (análise de requisitos), e ordenaranse estes por prioridade, dando lugar ao Backlog. Tamén se definirá a especificación de cada requisito e se deseñarán as probas que os verifiquen.

Desenvolvemento: Abrangue dende o Sprint 01 ata o Sprint 11, ambos inclusive (20 semanas en total). Nesta fase elaborárase o produto de acordo cos requisitos

Documentación: Abrangue 5 semanas de traballo. Nesta fase recompilárase toda a documentación xerada nas fases anteriores, e confeccionarase a memoria e máis a presentación, que constituirán os entregables do Traballo de Fin de Grao.

En total, o proxecto traballarase durante un período de 27 semanas (189 días, algo máis de 6 meses), co cal, para realizar as 401,25 horas de traballo necesarias teremos que levar un ritmo de traballo aproximado de 15,28 horas semanais (unha media de 2 horas e cuarto diarias). Non nos convén asumir un ritmo de

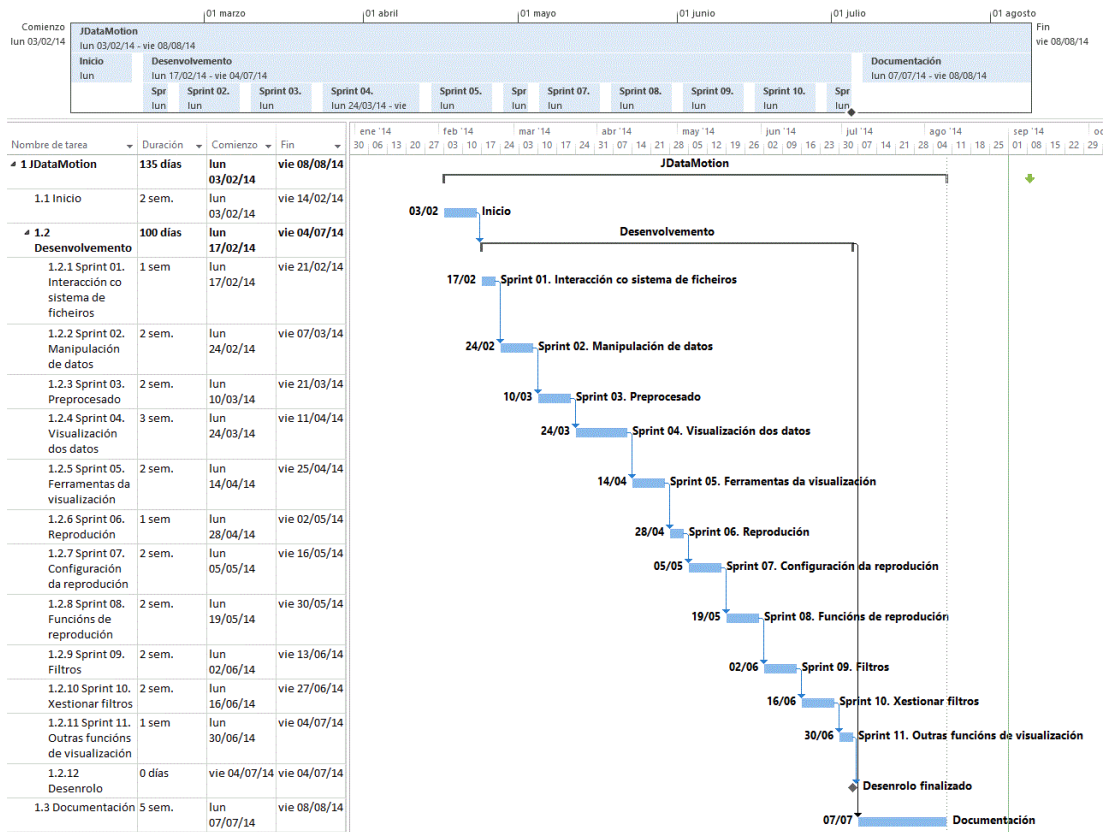


Figura 2.1: Diagrama de Gantt

traballo maior, pois durante ese período de tempo o alumno deberá repartir a súa axenda entre este proxecto, o resto de materias, as prácticas en empresa, etc.

A continuación e en base ao especificado no anteproxecto (sprints e planificación temporal dos mesmos), exporemos na Figura 2.1 o Diagrama de Gantt que ilustra a planificación temporal das fases. Cómpre destacar que a fase de Desenvolvemento será dividida nos sprints que a compoñen directamente.

Do mesmo xeito, coa referencia do anteproxecto extraemos na Figura 2.2 o Esquema de Descomposición do Traballo (EDT) do proxecto. Cada fase anterior amosarase dividida nas tarefas que a compoñen. Cabe destacar que na fase de Desenvolvemento, para cada sprint, as tarefas que ten asignadas son os propios requisitos funcionais (RFs) a implementar nese sprint, ou o que é o mesmo, a implementación de cada RF constitúe a tarefa que o representa. Por exemplo, o RF 'Insertar filtros' impleméntase na tarefa do mesmo nome: Insertar filtros.

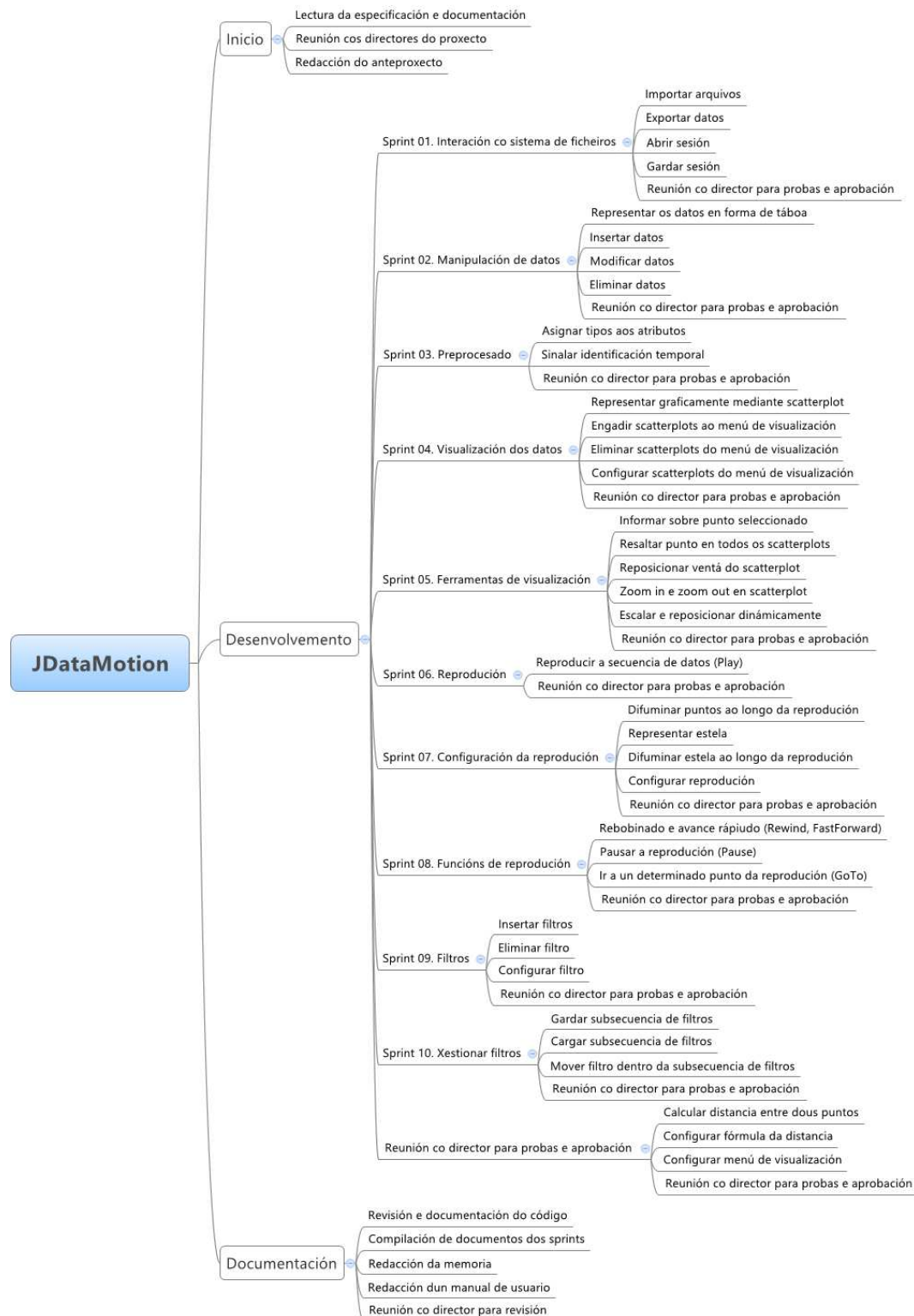


Figura 2.2: Esquema de Descomposición do Trabalho (EDT)

2.4. Xestión da configuración

Todo proxecto ten elementos de interese para incluír na xestión da configuración. Estes elementos caracterízanse porque son candidatos a sufrir cambios que poden ameazar o correcto desenvolvemento do proxecto. A xestión da configuración trata de manter a integridade do proxecto perante a estes cambios. O noso deber é identificar que obxectos do proxecto (sexan entregables ou resultados parciais do mesmo) merecen a súa inclusión na xestión da configuración, e por outra parte, temos que especificar que ferramentas empregaremos para dar soporte a esta característica.

Para este caso consideraremos ao código fonte do proxecto (e máis das súas probas) e á documentación como elementos de configuración, que se corresponden coas carpetas 'src', 'test' e 'doc' respectivamente. O código fonte é o sustento do noso proxecto, e os cambios no seu contido veranse directamente reflexados no produto a entregar, polo que é mester incluír este elemento na xestión da configuración. Tamén incluiremos o código fonte das probas porque debemos respectar a integridade entre estas e o propio proxecto. Por outra parte, a documentación sufrirá cambios de xeito paralelo ao código, e evolucionará da man deste ao longo da vida do proxecto (rexistrará a súa especificación de requisitos, o seu deseño, etc.), así que tamén debe ser un elemento a considerar en aras de preservar a integridade do proxecto. En resumo, faremos seguimento de cambios dos tres directorios ('src', 'test' e 'doc'), e para iso botaremos man do software GitHub[9].

GitHub é unha forxa para darlle aloxamento a distintos tipos de proxectos, por medio do sistema de control de versións Git. Para aloxar o noso proxecto crearemos un repositorio local e outro remoto, chamando a ambos 'JDataMotion', e outorgándolle ao remoto permisos de lectura e escritura para o alumno e permisos de lectura (ou de escritura tamén, opcionalmente) para os titores. Deste xeito estes poderán descargar a última versión do proxecto en calquera momento, mentres que o alumno poderá ir facendo '*pushes*' cada certo tempo cos cambios implementados.

O proxecto contén moitos máis elementos, pero non podemos consideralos a todos aptos para a xestión de configuración por diversos motivos: as librarías empregadas (directorio 'lib') non cambian (e no caso de querer actualizar algunha, asúmesese que non deben xurdir problemas de integridade gracias á compatibilidade entre versións), e os ficheiros de código obxecto e de distribución ('build' e 'dist') dependen directamente do código fonte (que xa é un elemento de configuración), pois xéranse como resultado da súa compilación.

2.5. Análise de custos

A estimación dos custos de desenvolvemento do proxecto amósase no Cadro 2.1. Para ela, consideramos a adquisición dun novo equipo informático. As horas de traballo neste caso non van ter un valor económico asociado, como consecuencia de que este proxecto pertenza a un Traballo de Fin de Grao, pois as horas do traballo do alumno correspóndense coas que este debe cumprimentar para a obtención do título. Para o consumo eléctrico tivemos en conta o prezo do kWh en España[5] e fixemos unha estimación[6] do consumo en Watios dun equipo informático, que a plena potencia pode traballar a 120 vatios. Considerando que o desenvolvemento do traballo durará 401.25 horas, necesitaremos $120 \text{ W} * 401.25 \text{ h} = 48150 \text{ Wh} = 48,15 \text{ kWh}$. Ademais, tratarase sempre de buscar solucións 'freeware' ou de software gratuíto ás distintas necesidades de módulos adicionais que vaian xurdindo, polo que o custo económico destas será nulo.

Activo	Cantidade	C.U. sen IVE	IVE	Custo total
Ordenador portátil	1	570,00 €	21 %	689,70 €
Horas de traballo	401,25 horas	0,00 €/hora	21 %	0,00 €
Consumo eléctrico	48,15 kWh	0,12 €/kWh	21 %	7,23 €
Total				696,93 €

Cadro 2.1: Custos

Capítulo 3

Análise de requisitos

A extracción dos requisitos dun proxecto é unha fase fundamental na realización de calquera proxecto, pois inflúe non só nas propias tarefas a desenvolver para a súa implementación, se non tamén na valoración do produto final e da súa calidade. A obtención de requisitos adóitase facer durante ou tras unha reunión co cliente.

3.1. Casos de uso

Os casos de uso empréganse para modelar e representar nun primeiro momento cómo se vai realizar a interacción entre o sistema e os usuarios del, tamén coñecidos como actores. Os casos de uso constitúen as posibilidades das que dispón cada actor. Este análise resulta especialmente útil en entornas orientadas a usuarios con distinta prioridade (un administrador, un usuario invitado, un usuario rexistrado, un usuario prémium, etc.) nas que cada un deses actores ten acceso a uns casos de uso específicos (por exemplo, moitas aplicacións web). Por tanto, a riqueza dos diagramas de casos de uso radica na variedade de tipos de usuario (actores). A nosa aplicación non necesita facer distinción algunha entre os tipos de usuario que poden facer uso dela. Todos van dispor das mesmas funcionalidades. En conclusión, o diagrama de casos de uso non aportaría ningunha información nova, así que será omitido.

3.2. Requisitos funcionais

RF01

Título

Importar arquivos con datos para o experimento

Descrición

A aplicación debe permitir cargar do sistema de arquivos un ficheiro que conteña unha secuencia de datos (nun formato axeitado segundo o RNF02) para ser utilizados no experimento.

Casos de uso relacionados**Importancia**

Esencial

RF02

Título

Exportar datos

Descrición

A aplicación debe permitir almacenar nun arquivo o conxunto de datos do arquivo actual (tendo en conta filtrados, modificacións, datos engadidos ou eliminados...). Os arquivos de saída deben respectar o RNF02.

Importancia

Esencial

RF03

Título

Gardar sesión

Descrición

A aplicación debe permitir gardar en disco a sesión (ou experimento) actual tal e como está no momento de executar esta acción.

Importancia

Esencial

RF04

Título

Abrir sesión

Descrición

A aplicación debe permitir restaurar unha sesión (ou experimento) gardada anteriormente, de xeito que se atope exactamente igual ca no momento en que se gardou.

Importancia

Esencial

RF05

Título

Representar os datos en forma de táboa

Descrición

A aplicación debe ser capaz de amosar os datos segundo unha táboa na que figuren cabeceiras, tipos, valores...

Importancia

Esencial

RF06

Título

Insertar datos no experimento actual

Descrición

A aplicación debe permitir a inserción dinámica de datos no experimento actual.

Importancia

Esencial

RF07

Título

Modificar datos no experimento actual

Descripción

A aplicación debe permitir a modificación dinámica de datos no experimento actual.

Importancia

Esencial

RF08**Título**

Eliminar datos no experimento actual

Descripción

A aplicación debe permitir a eliminación dinámica de datos no experimento actual.

Importancia

Esencial

RF09**Título**

Asignar tipos aos atributos dun arquivo importado

Descripción

A aplicación debe permitir especificar os tipos de atributos presentes no arquivo importado. Por exemplo, os datos cuantitativos poderían ser enteiros ou reais, mentras que os cualitativos serían algo distinto (mesmamente strings).

Importancia

Esencial

RF10**Título**

Sinalar identificación temporal

Descripción

A aplicación debe permitir sinalar unha columna que exprese o orde ou a temporalidade dunha tupla, ou ben definir esta columna manualmente.

Importancia

Esencial

RF11**Título**

Representar graficamente mediante scatterplot

Descripción

A aplicación debe ser capaz de representar gráficamente (mediante scatterplots) o conxunto de parámetros de entrada. Concretamente, débense poder representar ata 4 parámetros por cada scatterplot (ordeadas, abscisas, cor e forma dos puntos). A cor e a forma representan valores discretos, pero ademáis a forma pode representar valores continuos no caso dun degradado. Todos os scatterplots estarán englobados dentro do “menú de visualización”, que cumprirá co RNF06.

Importancia

Esencial

RF12**Título**

Engadir scatterplots ao menú de visualización

Descripción

A aplicación debe permitir engadir dinámicamente novos scatterplots dentro do menú de visualización.

Importancia

Esencial

RF13**Título**

Eliminar un scatterplot do menú de visualización

Descripción

A aplicación debe permitir eliminar un scatterplot do menú de visualización.

Importancia

Esencial

RF14**Título**

Configurar un scatterplot do menú de visualización

Descrición

A aplicación debe permitir especificar para cada scatterplot do menú de visualización a tupla de atributos a comparar. Tamén se debe poder elixir dende o eixo de representación para cada atributo como a cor ou forma dos puntos. Ademáis tense que dispoñer da opción especificar numéricamente a posición x_0 e y_0 na que comeza a ventá de visualización, e o ancho e alto desta ventá, o cal constitúe implícitamente un xeito de situar a ventá de visualización, de facer zoom sobre ela e no caso da relación ancho/alto, mesmo de establecer escalas distintas para cada eixo. Isto último podería omitirse, en beneficio dun comportamento dinámico e por defecto da ventá de visualización, que se adaptaría para englobar a todos os puntos representados.

Importancia

Esencial

RF15**Título**

Detallar punto seleccionado dentro do scatterplot

Descrición

Cada punto (non difuminado completamente) dos scatterplots pode ser seleccionado para ver nun apartado os seus detalles (todos os seus atributos, marca temporal...).

Importancia

Esencial

RF16**Título**

Resaltar punto en scatterplots

Descrición

Cada punto seleccionado dentro dun scatterplot resaltarase tanto nel coma en todos os demais scatterplots (que plasmarán outras proxeccións do mesmo punto).

Importancia

Esencial

RF17**Título**

Desprazar a ventá de visualización por arrastre de cada scatterplot (reposicionar)

Descrición

Para cada scatterplot poderemos usar unha ferramenta “man” para desprazar a ventá polo scatterplot.

Importancia

Esencial

RF18**Título**

Facer zoom in e zoom out en cada scatterplot (escalar)

Descrición

Para cada scatterplot poderemos usar unha ferramenta de Zoom in e outra de Zoom out para facer zoom do scatterplot. O zoom aumentará ou diminuírá a razón de X1.2

Importancia

Esencial

RF19**Título**

Escalar e reposicionar dinámicamente

Descrición

Para cada scatterplot poderemos seleccionar que a ventá de visualización que o enfoca se adapte dinámicamente ao conxunto de datos representados (movéndose, afastándose, aproximándose... para englobar todos os datos).

Importancia

Esencial

RF20**Título**

Reproducir a secuencia de datos (Play)

Descrición

A aplicación debe de permitir que a visualización dos scatterplots poida basarse na variable temporal (ou de orde) para reproducir a secuencia de datos, amosando os datos de cada scatterplot baixo unha secuencia de vídeo. Nesta secuencia engadiríase á visualización en cada instante a tupla de atributos asociada a esa marca temporal.

Importancia

Esencial

RF21**Título**

Difuminar puntos ao longo da reprodución

Descrición

A aplicación debe permitir difuminar os puntos xa representados a través do avance temporal.

Importancia

Esencial

RF22**Título**

Representar estela

Descrición

A aplicación debe de permitir que cada novo punto ploteado se ligue ao último representado no scatterplott por medio dunha liña recta.

Importancia

Esencial

RF23**Título**

Difuminar estela ao longo da reprodución

Descrición

A aplicación debe permitir difuminar as estelas xa representadas a través do avance temporal.

Importancia

Esencial

RF24**Título**

Configurar a reprodución da secuencia de datos

Descrición

A aplicación debe de permitir que a visualización dos scatterplots sexa configurable en canto a tempo transcurrido entre marcas temporais cando estas sexan de orde, que a velocidade do Play sexa x1, x2 ou x4 ou que se reproduza cara adiante ou cara atrás. Ademáis débese poder especificar o número de marcas temporais que durará o difuminado dos puntos que se ploteen, de xeito que durante ese intervalo cada punto se vaia difuminando ata desaparecer. Pode ser 0 para que os puntos non se difuminen. A aplicación tamén debe permitir especificar o número de marcas temporais que durará o difuminado das estelas que se ploteen, de xeito que durante ese intervalo cada estela xa debuxada se vaia difuminando ata desaparecer. Pode ser 0 para que as estelas non se difuminen.

Importancia

Esencial

RF25**Título**

Rebobinado e avance rápido da reprodución (Rewind, FastForward)

Descrición

A aplicación debe permitir avanzar e retroceder a alta velocidade (X8) a reprodución.

Importancia

Esencial

RF26**Título**

Pausar a reprodución (Pause)

Descrición

A aplicación debe permitir parar a reprodución na marca de tempo na que se atope ao executar esta acción, mantendo as visualizacións para ese momento.

Importancia

Esencial

RF27**Título**

Ir a un determinado instante dentro do intervalo temporal da reprodución (GoTo)

Descrición

A aplicación debe permitir situarse directamente sobre un instante de tempo, mantendo a reprodución pausada sobre esa marca temporal, e visualizando os scatterplots tal e como deben estar nese momento.

Importancia

Esencial

RF28**Título**

Insertar filtros para os datos do experimento

Descrición

A aplicación debe permitir engadir unha serie de filtros que se aplicarán de xeito secuencial sobre a secuencia de datos coa que se esté a traballar. Chamáremoslle "secuencia de filtros."^a esta secuencia.

Importancia

Esencial

RF29**Título**

Eliminar un filtro para os datos do experimento

Descrición

A aplicación debe permitir eliminar un determinado filtro dentro da secuencia de filtros.

Importancia

Esencial

RF30**Título**

Configurar filtros para os datos do experimento

Descrición

A aplicación debe permitir seleccionar un determinado filtro dentro da secuencia de filtros para modificar a regra de filtrado implícita.

Importancia

Esencial

RF31**Título**

Gardar unha subsecuencia de filtros do experimento

Descrición

A aplicación debe permitir gardar unha subsecuencia de filtros dentro dos que se estén aplicando sobre o experimento. Esta subsecuencia pode comprender tanto un só filtro como a secuencia de filtros enteira.

Importancia

Esencial

RF32**Título**

Cargar unha secuencia de filtros para o experimento

Descrición

A aplicación debe permitir cargar do sistema de arquivos unha secuencia de filtros que se engadirá á cabeza da secuencia de filtros (a cal pode estar vacía). Esta secuencia tamén pode estar composta por un só filtro.

Importancia

Esencial

RF33**Título**

Mover os filtros dentro da secuencia de filtros

Descrición

A aplicación debe permitir desprazar un filtro dentro da secuencia de filtros do experimento, de xeito que o orde de aplicación dos filtros varíe. O desprazamento realizarase insertando o filtro en cuestión nunha nova posición.

Importancia

Esencial

RF34**Título**

Calcular distancia entre dous puntos do plano

Descrición

A aplicación debe permitir o cálculo da distancia entre dous puntos do plano.

Importancia

Esencial

RF35**Título**

Configurar a fórmula para achar distancia entre dous puntos do plano

Descrición

A aplicación debe permitir a introdución da fórmula que se desexe para calcular a distancia entre dous puntos

Importancia

Esencial

RF36

Título

Configurar o menú de visualización

Descrición

A aplicación debe permitir cambiar os parámetros de visualización dos scatterplots que compoñen o menú de visualización, por exemplo, a cor das estelas, do fondo, dos eixos... ou a fonte, tamaño de letra...

Importancia

Optativa

3.3. Requisitos de calidade

RC01

Título

Latencia mínima para o procesamento

Descrición

A aplicación debe responder nun tempo razoable ás operacións executadas polo usuario, e intentar que esa latencia escale de xeito controlado ao aumentar a talla dos parámetros.

Importancia

Esencial

3.4. Requisitos de deseño

RD01

Título

Modularidade no deseño dos filtros

Descrición

A aplicación debe facilitar unha interface para a inclusión e uso de filtros

personalizados por parte de calquera desenvolvedor de software que a implemente dentro do proxecto.

Importancia

Esencial

3.5. Requisitos non funcionais

RNF01

Título

Formatos de entrada admitidos ao importar e exportar arquivos

Descrición

A aplicación debe estar preparada para importar e exportar arquivos en distintos formatos, como son o CSV e ARFF.

Importancia

Esencial

RNF02

Título

Relación programa-sesión

Descrición

Cada instancia do programa debe traballar cunha única sesión (experimento).

Importancia

Esencial

RNF03

Título

Representación matricial dos scatterplots

Descrición

Os scatterplots represéntanse de xeito matricial, facendo que cada parámetro dentro dun eixo sexa enfrentado a cada un dos demáis do outro eixo, e en cada punto desa dupla se sitúe o scatterplot que compara ambos parámetros.

Deste xeito, os scatterplots non son acumulables: se temos un que representa X fronte a Y, non podemos engadir outro que represente X fronte a Y, pois ocuparían ambos a mesma cela dentro da matriz de scatterplots.

Importancia

Esencial

RNF04**Título**

Entrega dentro de prazo

Descrición

Débese entregar unha versión funcional e documentada antes do día 8 de Setembro de 2014, ás 14:00 horas, pois é o momento no que remata o prazo de entrega.

Importancia

Esencial

3.6. RFS dos sprints

Aínda que xa foron especificados para o Esquema de Descomposición do Traballo (EDT) do apartado de planificación (véxase Figura 2.2), imos a detallar a asignación de requisitos funcionais (RFS) aos distintos sprints. Cómpre lembrar que para o EDT, en cada sprint da fase de Desenvolvemento e Inicio considerábase como tarefas aos requisitos funcionais a implementar nela.

Sprint 01

Nome: Interacción co sistema de ficheiros

Fase: Desenvolvemento

Comezo: 17/02/2014

Finalización: 24/02/2014

RFS a implementar: RF01, RF02, RF03, RF04

Sprint 02

Nome: Manipulación de datos

Fase: Desenvolvemento

Comezo: 24/02/2014

Finalización: 10/03/2014

RFs a implementar: RF05, RF06, RF07, RF08

Sprint 03

Nome: Preprocesado

Fase: Desenvolvemento

Comezo: 10/03/2014

Finalización: 24/03/2014

RFs a implementar: RF09, RF10

Sprint 04

Nome: Visualización dos datos

Fase: Desenvolvemento

Comezo: 24/03/2014

Finalización: 14/04/2014

RFs a implementar: RF11, RF12, RF13, RF14

Sprint 05

Nome: Ferramentas de visualización

Fase: Desenvolvemento

Comezo: 14/04/2014

Finalización: 28/04/2014

RFs a implementar: RF15, RF16, RF17, RF18, RF19

Sprint 06

Nome: Reprodución

Fase: Desenvolvemento

Comezo: 28/04/2014

Finalización: 05/05/2014

RFs a implementar: RF20

Sprint 07

Nome: Configuración da reprodución

Fase: Desenvolvemento

Comezo: 05/05/2014

Finalización: 19/05/2014

RFs a implementar: RF21, RF22, RF23, RF24

Sprint 08

Nome: Funcións de reprodución

Fase: Desenvolvemento

Comezo: 19/05/2014

Finalización: 02/06/2014

RFs a implementar: RF25, RF26, RF27

Sprint 09

Nome: Filtros

Fase: Desenvolvemento

Comezo: 02/06/2014

Finalización: 16/06/2014

RFs a implementar: RF28, RF29, RF30

Sprint 10

Nome: Xestionar filtros

Fase: Desenvolvemento

Comezo: 16/06/2014

Finalización: 30/06/2014

RFs a implementar: RF31, RF32, RF33

Sprint 11

Nome: Outras funcións de visualización

Fase: Desenvolvemento

Comezo: 30/06/2014

Finalización: 07/07/2014

RFs a implementar: RF34, RF35, RF36

Capítulo 4

Deseño

Deseño: cómo se realiza o Sistema, a división deste en diferentes compoñentes e a comunicación entre eles. Así mesmo, determinarase o equipamento hardware e software necesario, xustificando a súa elección no caso de que non fora un requisito previo. Debe achegarse a un nivel suficiente de detalle que permita comprender a totalidade da estrutura do produto desenvolvido, utilizando no posible representacións gráficas.

Capítulo 5

Exemplos

5.1. Un exemplo de sección

Esta é *letra cursiva*, esta é **letra negrilla**, esta é letra subrallada, e esta é **letra curier**. Letra tiny, scriptsize, small, large, Large, LARGE e moitas más. Exemplo de fórmula: $a = \int_0^\infty f(t)dt$. E agora unha ecuación aparte:

$$S = \sum_{i=0}^{N-1} a_i^2. \quad (5.1)$$

As ecuaciones se poden referenciar: ecuación (5.1).

5.1.1. Un exemplo de subsección

O texto vai aquí.

5.1.2. Outro exemplo de subsección

O texto vai aquí.

Un exemplo de subsubsección

O texto vai aquí.

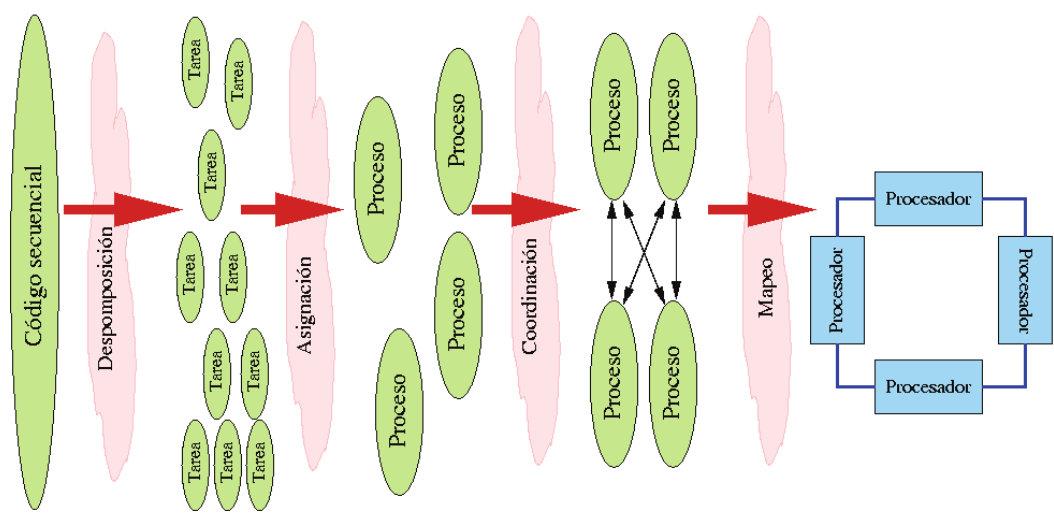


Figura 5.1: Esta é a figura de tal e cal.

Izquierda	Derecha	Centrado
ll	r	cccc
llll	rrr	c

Cadro 5.1: Esta é a táboa de tal e cal.

Un exemplo de subsubsección

O texto vai aquí.

Un exemplo de subsubsección

O texto vai aquí.

5.2. Exemplos de figuras e cadros

A figura número 5.1.

O cadro (taboa) número 5.1.

5.3. Exemplos de referencias á bibliografía

Este é un exemplo de referencia a un documento descargado da web [?]. E este é un exemplo de referencia a unha páxina da wikipedia [?]. Agora un libro [?] e agora unha referencia a un artigo dunha revista [?]. Tamén se poden por varias referencias á vez [?, ?].

5.4. Exemplos de enumeracións

Con puntos:

- Un.
- Dous.
- Tres.

Con números:

1. Catro.
2. Cinco.
3. Seis.

Exemplo de texto verbatim:

```
0 texto          verbatim
  se visualiza tal
    como se escribe
```

Exemplo de código C:

```
#include <math.h>
main()
{  int i, j, a[10];
   for(i=0;i<=10;i++) a[i]=i; // comentario 1
   if(a[1]==0) j=1; /* comentario 2 */
   else j=2;
}
```

Exemplo de código Java:

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

}

Capítulo 6

Conclusións e posibles ampliacións

Conclusións e posibles ampliacións

Apéndice A

Manuais técnicos

Manuais técnicos: en función do tipo de Traballo e metodoloxía empregada, o contido poderase dividir en varios documentos. En todo caso, neles incluírase toda a información precisa para aquelas persoas que se vaian a encargar do desenvolvemento e/ou modificación do Sistema (por exemplo código fonte, recursos necesarios, operacións necesarias para modificacións e probas, posibles problemas, etc.). O código fonte poderase entregar en soporte informático en formatos PDF ou postscript.

Apéndice B

Manuais de usuario

Manuais de usuario: incluírán toda a información precisa para aquelas persoas que utilicen o Sistema: instalación, utilización, configuración, mensaxes de erro, etc. A documentación do usuario debe ser autocontida, é dicir, para o seu entendemento o usuario final non debe precisar da lectura de outro manual técnico.

Apéndice C

Licenza

Se se quere pór unha licenza (GNU GPL, Creative Commons, etc), o texto da licenza vai aquí.

Bibliografía

- [1] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. Sitio web <http://www.cs.waikato.ac.nz/ml/weka/>.
- [2] Proxecto JFreeChart. Sitio web <http://www.jfree.org/jfreechart>.
- [3] Formato de Archivo Atributo-Relación (ARFF). Información disponible en <http://www.cs.waikato.ac.nz/ml/weka/arff.html>.
- [4] Introduction to JFreeChart. Sitio web <http://www.codeproject.com/Articles/650480/Introduction-to-JFreeChart>.
- [5] Tarifasgasluz. Precio del kWh en España. Sitio web <http://tarifasgasluz.com/faq/precio-kwh/espana>.
- [6] Electricity usage of a Laptop or Notebook. Sitio web http://energyusecalculator.com/electricity_laptop.htm.
- [7] Scrum. Definición extraída de <http://es.wikipedia.org/wiki/Scrum>.
- [8] Acunote. Sitio web <http://www.acunote.com/>.
- [9] GitHub Sitio web <https://github.com/>.