

Solicitud de entrega de anteproxecto de TFG

Titulación: Grao en Enxeñaría Informática

Datos do/a Alumno/a			
Nome: Abel Fernández Nandín		DNI: 53186810T	
Enderezo: Avenida Camelias 117 - E - 1ºD		Localidade: Vigo	
Provincia: Pontevedra	C.P.: 36211	Teléfono: 663408523	E-mail: afnandin@esei.uvigo.es

Datos do Traballo de Fin de Grao TFG	
Título do TFG: Asistente de Laboratorio para MALDI	
Titor/a do TFG: Miguel Reboiro Jato	DNI: 44477792R
Cotitor/a do TFG (se procede):	DNI:
Área de coñecemento: Linguaxes e Sistemas Informáticos	
Departamento: Informática	

☐ Marque esta opción se o TFG se vai desenvolver nunha empresa.

En caso afirmativo, indique o nome da empresa:

¹Código de TFG:

Ourense, 20 de marzo de 2014

Recibín o código de TFG asignado, o/a
alumno/a

Vº e Pr. O/A titor/a do TFG

Asdo: Abel Fernández Nandín

Asdo: Miguel Reboiro Jato

Copia para o solicitante¹ A cubrir pola administración do Centro

Solicitud de entrega de anteproxecto

Titulación: Grao en Enxeñaría Informática

Datos do/a Alumno/a			
Nome: Abel Fernández Nandín		DNI: 53186810T	
Enderezo: Avenida Camelias 117 - E - 1º D		Localidade: Vigo	
Provincia: Pontevedra	C.P.: 36211	Teléfono: 663408523	E-mail: afnandin@esei.uvigo.es

Datos do Traballo de Fin de Grao TFG	
Título do TFG: Asistente de Laboratorio para MALDI	
Titor/a do TFG: Miguel Reboiro Jato	DNI: 44477792R
Cotitor/a do TFG (se procede):	DNI:
Área de coñecemento: Linguaxes e Sistemas Informáticos	
Departamento: Informática	

☐ Marque esta opción se o TFG se vai desenvolver nunha empresa.

En caso afirmativo, indique o nome da empresa:

1Código de TFG:

Ourense, 20 de marzo de 2014

Recibín o código de TFG asignado, o/a
alumno/a

Vº e Pr. O/A Director/a do TFG

Asdo: Abel Fernández Nandín

Asdo: Miguel Reboiro Jato

DESCRIPCIÓN DO ANTEPROXECTO

(Adapta-lo tamaño do documento según sexa necesario. Consulta-lo art. 9 do Regulamento de TFG da ESEI)

TÍTULO: Asistente de laboratorio para MALDI

Introdución

(incluir a presentación do problema plantexado)

Nos últimos anos a investigación en bioloxía sufriu unha importante revolución coa chegada da xenómica e da proteómica, que supuxeron o paso de facer investigacións sobre pequenos conxuntos de xenes ou proteínas a estudar miles destes compoñentes biolóxicos de forma simultánea, chegando incluso a facer estudos de xenomas ou proteomas completos. Mentres que na xenómica nos atopamos cos microarrays de ADN e, máis recentemente, cos estudos de secuenciación masiva ou *Next Generation Sequencing* (NGS) como máximos expoñentes das técnicas de análise, na proteómica a técnica de maior relevancia é a espectrometría de masas.

Unha das técnicas máis destacadas de espectrometría de masas para proteómica é a denominada *Matrix-Assisted Laser Desorption/Ionization* (MALDI), especialmente na versión Time-Of-Flight (MALDI-TOF). Esta é unha técnica de ionización suave para a identificación e cuantificación de moléculas mediante a medición do seu tempo de voo a través dun campo electromagnético despois de seren ionizadas. Esta técnica resulta de especial interese pola súa capacidade para analizar biomoléculas, habitualmente proteínas e péptidos, e moléculas orgánicas de maior tamaño. Ademais, destaca pola posibilidade de analizar decenas de mostras das que se obteñen medicións de centos de moléculas nunha única análise.

Un problema común á xenómica, proteómica e resto de ciencias ómicas é a gran cantidade de información xerada nunha única análise que dificulta a súa posterior análise e manexo. A única forma viable de realizar estas tarefas e mediante a aplicación de técnicas informáticas. Esta dependencia viuse reflectida nos últimos anos no importante auxe do campo da bioinformática.

Este traballo trata sobre a creación dunha ferramenta bioinformática de asistencia a investigadores durante o proceso de deposición de mostras en placas para a análise por MALDI-TOF. Este é un proceso que, habitualmente, consume unha gran cantidade de tempo, non só polo feito de ter que deposicionar unha gran cantidade de mostras, senón porque previamente é preciso rexistrar a posición de cada unha das mostras dentro da placa MALDI. Este proceso, que normalmente se fai de forma manual nun modelo en papel, serve para poder identificar a que mostra corresponde cada un dos ficheiros de resultados xerados pola análise, que tamén deberán ser ordenados de forma manual. Ademais, o feito de ter que facer este traballo manualmente leva a que os investigadores deposicionen as mostras de forma ordenada segundo a súa condición, algo que pode derivar na aparición de desviacións non desexadas nos resultados. Polo tanto, a posibilidade de automatizar tanto o proceso de creación de modelos de deposición de mostras, como a posterior organización dos ficheiros de resultados, preséntase de gran interese para os investigadores pola potencial redución do tempo necesario para preparar unha mostra e do número de erros de orixe humano.

Obxectivos

(solución proposta, especificación funcional, ...)

O principal obxectivo deste proxecto é o desenvolvemento dunha aplicación que permita deseñar como se fará a deposición de mostras dun experimento en placas MALDI e que facilite organización posterior dos resultados da análise.

Para acadar este obxectivo principal cómpre superar os seguintes obxectivos:

- Permitir deseñar a forma na que se fará a deposición de mostras en placas MALDI. Este proceso debe resultar sinxelo e intuitivo para o usuario.
- Permitir a ordenación de mostras de xeito manual ou de forma automática baixo algún criterio determinado.
- Proporcionar un asistente que facilite a labor de deposición de mostras en placas MALDI no laboratorio segundo un deseño previo.
- Reorganizar os arquivos xerados polo MALDI para a súa análise posterior con ferramentas bioinformáticas.

Descrición técnica

(arquitectura técnica, tecnoloxías, linguaxes de programación, ...)

A aplicación que se desexa construír estará implementada sobre ZK, un *framework* para o desenvolvemento de aplicacións web en Java baseado en AJAX. A arquitectura ZK consta de dúas partes, cliente e servidor, tal e como se indica na *Figura 1*. A parte do cliente está formada polo navegador cliente, na que os usuarios interactúan cos *widgets* propios de ZK, que se comunican co motor cliente de ZK. Este motor no cliente realiza peticións (e recibe respostas) AJAX ao servidor. Estas peticións son atendidas no servidor polo motor de actualización, que se comunica cos diferentes compoñentes de ZK para engadir eventos á cola de eventos. Deste xeito conséguese que manter o estado dos compoñentes sincronizado entre o cliente e o servidor.

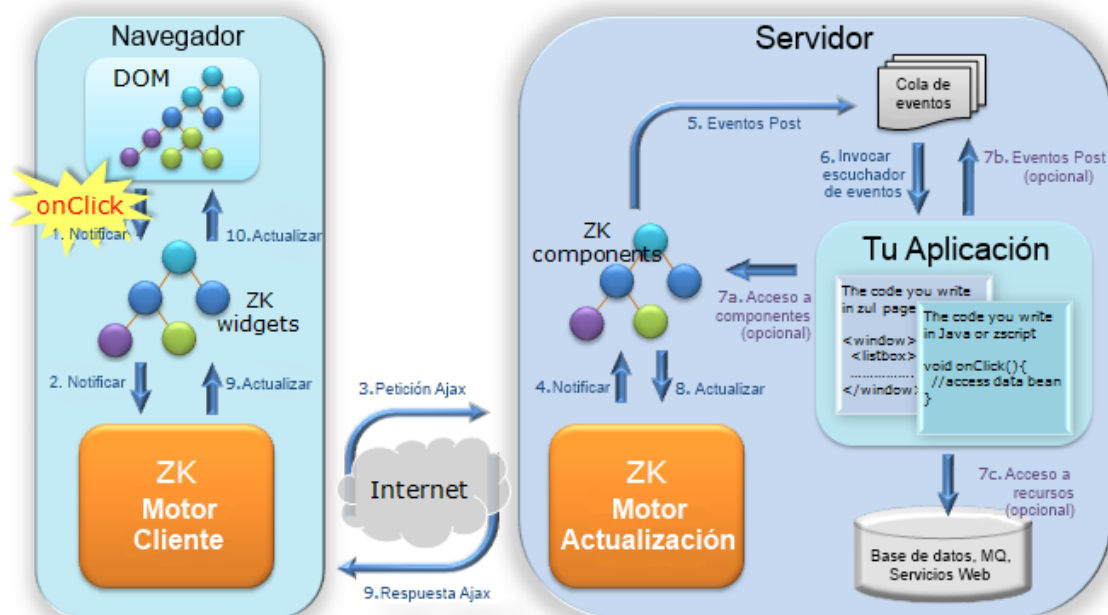


Figura 1: Arquitectura ZK

Unha aplicación ZK pode acceder aos recursos do seu servidor, formar unha interface de usuario con compoñentes, escoitar as actividades do usuario, e despois manipular os compoñentes para actualizar a interface. Todas estas actividades pódense levar a cabo no servidor, e a sincronización do estado dos compoñentes entre o navegador cliente e o servidor é realizada por ZK automaticamente de forma transparente para o desenvolvedor.

Ao executarse no servidor, unha aplicación ZK ten acceso á pila completa de tecnoloxías Java habituais nos servidores de aplicacións. As tecnoloxías propias do usuario, como AJAX ou Push, son abstraídas en obxectos de evento. Por outra parte, a interface de usuario está composta de compoñentes POJO (*Plain Old Java Objects*), que fan que o seu manexo resulte sinxelo.

Proceso de desenvolvemento a empregar no proxecto

(débesse describir cada etapa)

Para o proceso de implementación da aplicación empregárase a metodoloxía de desenvolvemento áxil Scrum adaptada ás restricións impostas pola normativa de TFGs á que está suxeito este traballo.

A decisión de empregar unha metodoloxías áxiles vén motivada por estar estas centradas na optimización do tempo de desenvolvemento, maximizando a labor creativa e minimizando aquelas tarefas que non aportan valor ao proxecto. Por outra banda, escolleuse Scrum por ser unha das metodoloxías áxiles máis utilizadas no ámbito empresarial, no que a diminución do tempo para a consecuente redución de custes é moi importante. Scrum ofrece, ademais, un marco de traballo sumamente flexible, algo que resulta de interese no presente traballo por ter este un perfil parcialmente exploratorio.

A Metodoloxía Scrum

Scrum é unha metodoloxía áxil que está dirixida polo Backlog de Produto, o cal consiste nunha lista priorizada e completa de funcionalidades que se deben desenvolver para obter un produto. O feito de que a priorización sexa feita segundo o criterio do cliente fai que as características máis valiosas para este se desenvolvan en primeiro lugar. Por outra banda, é unha metodoloxía de desenvolvemento iterativa e incremental, na que cada iteración se desenvolve nun prazo de tempo fixo, denominado *sprint*, no que se implementan funcionalidades completas. O conxunto das funcionalidades desenvolvidas durante unha iteración constitúen, ao final da mesma, un incremento que dá lugar a un produto potencialmente publicable.

Como se comentou máis arriba, será preciso aplicar certas adaptacións a esta metodoloxía para ter en conta as restricións impostas pola normativa á que está suxeito este traballo. A restrición máis importante é o feito de que este traballo ten carácter individual, xa que a metodoloxía Scrum está orientada a equipos compostos, normalmente, por entre 5 e 9 membros. Por este motivo, adoptaranse as seguintes adaptacións:

- O equipo estará formado polo autor e o director deste traballo. Sendo o primeiro o responsable da parte relacionada co desenvolvemento do mesmo e o segundo da definición dos requisitos, actuando como cliente.
- As reunións diarias pasarán a realizarse dúas veces á semana. Posto que o obxectivo das mesmas é que todo o equipo de desenvolvemento estea ao tanto do traballo realizado por todos os membros, nun equipo tan reducido o posible beneficio vese diminuído e, polo tanto, a realización diaria suporía un custo de tempo excesivo.

Planificación do traballo e estimación temporal

(deberase incluír en tódolos casos un diagramas de Gantt , pódese entregar como anexo a esta documentación)

En Scrum asúmese a dificultade de realizar unha planificación realista ao comezo do proxecto, polo que se evita que esta sexa moi detallada. Pola contra, a planificación do proxecto distribúese ao longo do mesmo en varias reunións de planificación nas que participa todo o equipo. Estas reunións, principalmente, resúmense en reunións de planificación de publicación (*release*), na que se priorizan e seleccionan as funcionalidades que formarán parte da próxima publicación do sistema, e as reunións de planificación de *sprint*, nas que se deciden as funcionalidades a desenvolver no *sprint*. As reunións de publicación sóense celebrar cada tres ou seis meses, mentres que as de *sprint* se realizan ao comezo de cada iteración.

É importante ter en conta que na planificación de Scrum non se separan as distintas etapas de desenvolvemento de funcionalidades (p.ex. análise, deseño, implementación, etc.) como é habitual noutras metodoloxías. Pola contra, prefírese eliminar esta separación para favorecer que, por exemplo, o deseño, implementación e probas se realicen de forma conxunta e sinérxica.

Neste traballo considérase que a primeira publicación do sistema terá lugar cando finalice o mesmo e que, polo tanto, todos os *sprints* realizados estarán asociados a esta publicación. En base a isto, as tarefas a realizar no presente proxecto por orde de planificación son as seguintes:

1. Modelado do negocio no que se crearán as historias de usuario e dos criterios de aceptación das mesmas.
2. Investigación das posibles tecnoloxías a utilizar.
3. Planificación da publicación, na que se fará a estimación e priorización das historias de usuario.
4. O primeiro *sprint*.
5. O segundo *sprint*.
6. O terceiro e último *sprint*.

A elaboración da documentación realizarase durante toda a duración do proxecto. A duración estimada total é de 300 horas, nas que se dedicarán 20 horas semanais, dando lugar a 15 semanas de desenvolvemento. Co fin de cumprir estas 20 horas semanais, dedícaranse 4 horas ao día, 5 días a semana.

Dedicación semanal prevista (en horas/semana): 20	
Fase	Estimación temporal (en semanas)
Modelado de negocio	1
Investigación das posibles tecnoloxías a utilizar	1
Planificación de publicación	1
Sprint 1	4
Sprint 2	4
Sprint 3	4
TOTAL PROXECTO	15

<input type="checkbox"/> Proxecto	75d	17/02/2014	30/05/2014
Modelado de negocio	1s	17/02/2014	21/02/2014
Investigación das posibles tecnoloxías a utilizar	1s	24/02/2014	28/02/2014
Planificación da publicación	1s	03/03/2014	07/03/2014
Sprint 1	4s	10/03/2014	04/04/2014
Sprint 2	4s	07/04/2014	02/05/2014
Sprint 3	4s	05/05/2014	30/05/2014
<input type="checkbox"/> Documentación	75d	17/02/2014	30/05/2014
Manual técnico	15s	17/02/2014	30/05/2014
Manual de usuario	1s	26/05/2014	30/05/2014
Memoria	1s	26/05/2014	30/05/2014

Figura 2: Tarefas do proxecto

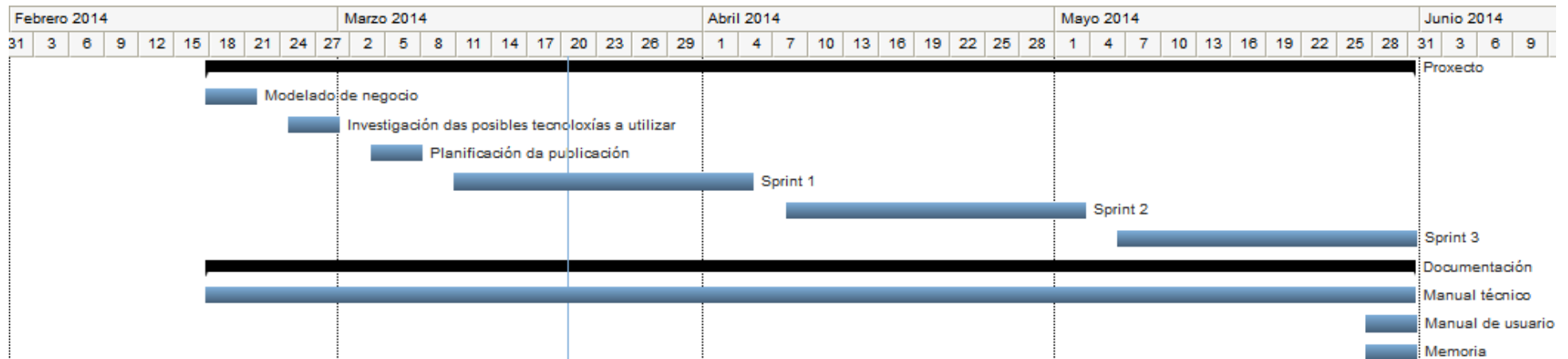


Figura 3: Diagrama de Gantt

Descrición da documentación a entregar no proxecto

Os documentos a entregar serán os seguintes:

- **Manual de usuario:** Documento informal orientado ao usuario no que se explicará como facer uso da ferramenta desenvolvida. Ademais, incluírá as instrucións necesarias para poder facer a instalación do sistema.
- **Manual técnico:** Documento formal que incluírá información puramente técnica referente á aplicación desenvolvida, co fin de facilitar as futuras tarefas de mantemento do sistema.
- **Memoria:** Documento formal no que se detalla todo o proceso de desenvolvemento do sistema, incluíndo as desviacións con respecto á planificación temporal e outros contratempos que puideran ter ocorrido.

Medios materiais necesarios

Hardware necesario:

- Ordenador portátil con acceso a internet e co software necesario instalado.

Software necesario:

- Sistema operativo (Microsoft Windows 8).
- Procesador de textos (Microsoft Word).
- Ferramenta para a planificación temporal do proxecto (Ganttter).
- Ferramenta para o modelado do proxecto en UML (Visual Paradigm).
- Entorno de desenvolvemento integrado para Java (Eclipse para Java EE).
- ZK Framework 7.0.
- ZK Studio Plugin para Eclipse.
- Servidor Web con capacidades de contedor de Servlet (Apache Tomcat 7).
- Sistema de control de versións (Git).

Bibliografía

La Guía de Scrum

Ken Schwaber y Jeff Sutherland

<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-ES.pdf#zoom=100>

Scrum Shortcuts without Cutting Corners: Agile Tactics, Tools, & Tips

Ilan Goldstein

Addison-Wesley, 2013

ZK Framework Manual

Potix Corporation

http://books.zkoss.org/wiki/ZK_Developer%27s_Reference

Observacións