

INGENIERÍA WEB
PROYECTO DE DESARROLLO
Ingeniería Informática
Universidad de Cádiz
Curso 2019-20

Profesorado de Ingeniería Web

10 de octubre de 2019

Índice

1. Introducción	2
2. Descripción del proyecto	2
2.1. Objetivos del sistema	2
2.2. Requisitos de tecnología	3
3. Método de desarrollo	3
3.1. Actividades	3
3.1.1. Inicio del proyecto	4
3.1.2. Desarrollo del proyecto	4
3.1.3. Cierre del proyecto	5
3.2. Roles	5
3.3. Herramientas	5
3.4. Entregables	6
4. Entrega y evaluación	7
4.1. Hitos de entrega	7
4.2. Evaluación	7

1. Introducción

El proyecto consiste en el desarrollo de una aplicación web a escala usando los métodos, técnicas y herramientas de la Ingeniería Web.

En el proyecto se pondrán en práctica las técnicas y métodos introducidos durante las clases teóricas de la asignatura. Asimismo, es fundamental el aprendizaje de los lenguajes, frameworks y herramientas necesarias para construir el proyecto, por lo que tal aprendizaje deberá realizarse como investigación individual de cada alumno y/o de cada equipo de proyecto.

El trabajo estará organizado en equipos de cuatro personas, donde cada una de ellas debe desempeñar varios roles. El trabajo se organizará en iteraciones sucesivas, cada una de las cuales podrá ser evaluada por el profesor y por los propios miembros del equipo. Excepcionalmente, en el caso de grupos con un menor número de estudiantes, se ajustará la carga de trabajo exigida por el profesor.

2. Descripción del proyecto



Este apartado es una especificación de los objetivos del sistema. Debe tomarse de partida para la definición de los requisitos funcionales, no funcionales y de información de común acuerdo con el cliente. Para simular una situación más próxima a la realidad, los requisitos podrán verse sujetos a cambio por parte del cliente (i.e. el profesor) tras las primeras evaluaciones que éste realice.

Una importante naviera especialista en cruceros de lujo por todo el mundo decide invertir en el desarrollo de un nuevo software empresarial. La aplicación web a construir deberá dar soporte a toda su flota de barcos. El objetivo de negocio no es otro que el de mejorar la experiencia y satisfacción del cliente durante sus vacaciones.

2.1. Objetivos del sistema

El sistema a desarrollar pretende satisfacer los siguientes objetivos:

Proporcionar un punto de información contextual para los clientes

Los huéspedes podrán consultar en cada momento información relativa a sus vacaciones en el barco: consejos para los viajeros, servicios e instalaciones disponibles en el barco, planos de localización, etc. Asimismo, el sistema proporcionará información específica de cada una de las escalas (horarios de llegada y salida, información de la ciudad, información meteorológica, consejos y excursiones programadas).

Facilitar el proceso de reservas y compras a bordo del barco

Los huéspedes podrán reservar mesa en los restaurantes de los que dispone el barco y en las actividades con aforo limitado. Además podrán contratar las excursiones a realizar en las diferentes escalas así como cualquier otro servicio adicional (tratamientos de belleza y SPA, asistencia médica, etc.). En cualquier momento los huéspedes podrán acceder a su cuenta de usuario desde donde podrán consultar los gastos realizados desde la aplicación web o desde cualquier otro punto de venta en el barco, como los de las tiendas de regalos o de los bares.

Proporcionar un back-office para la gestión de la información del sistema

Los usuarios autorizados podrán gestionar de forma cómoda todos los datos relativos a los barcos, itinerarios, escalas, excursiones, restaurantes, actividades, servicios disponibles, etc.

Monitorizar la actividad de la naviera

Los gerentes de la empresa podrán acceder en cualquier instante a indicadores claves del negocio, como por ejemplo el volumen total de facturación, excursiones más demandadas, actividades menos solicitadas, etc.

2.2. Requisitos de tecnología

Para la construcción del proyecto se usarán tecnologías *Java Enterprise Edition* (JEE) basadas en los frameworks *Spring Framework* y *Vaadin* para el desarrollo de aplicaciones web empresariales. Se hará uso de *Spring Boot* para la creación de una aplicación web auto-arrancable. Asimismo, se empleará alguna versión reciente del motor de bases de datos relacional *MySQL* para la persistencia de información.

3. Método de desarrollo

El desarrollo del proyecto se realizará con el [método SCRUM](#), un método ágil de desarrollo, que estará dividido en varias *iteraciones* semanales. En cada iteración se pretende alcanzar un objetivo mediante la realización de unas *actividades* encaminadas a aprender una serie de *técnicas* y empleando unas *herramientas* de ayuda al desarrollo. En estas actividades participarán una serie de *roles* y se generarán unos *entregables* en forma de software y de documentos.

3.1. Actividades

Los trabajos comenzarán con una fase de inicio en donde se planificará todo lo referente al proyecto, así como se prepararán todos los entornos necesarios. Luego comenzará el desarrollo en sí en donde todos los roles establecidos desempeñarán las labores propias de la Ingeniería Web en cada una de las iteraciones del proyecto. De manera simultánea, se realizarán diversas actividades de gestión y

evaluación del proyecto conforme el software va evolucionando. Finalmente en la fase de cierre, se acabarán las actividades necesarias y se procederá a la evaluación final.

3.1.1. Inicio del proyecto

Durante la primera fase del proyecto se requiere llevar a cabo una serie de actividades de planificación y configuración:

- Formar los equipos de trabajo
- Preparar el entorno de trabajo colaborativo: sistema de control de versiones de código, seguimiento del proyecto, repositorio documental, comunicaciones, etc.
- Desarrollar el plan de proyecto, incluyendo los roles asignados a cada miembro del equipo, sprints (hitos), entregables, periodicidad de reuniones, acuerdos relativos a la gestión del trabajo en grupo y justificación de las herramientas a utilizar durante el proyecto.
- Preparar los entornos de trabajo en local: IDE, base de datos, servidor de aplicaciones, etc.

3.1.2. Desarrollo del proyecto

Durante esta fase del proyecto se realizarán una serie de actividades tanto de desarrollo como de gestión y evaluación, enmarcadas dentro de iteraciones o incrementos. Son las siguientes:

- Planificar las historias de usuario necesarias para la nueva interacción (sprint backlog) como tareas en la herramienta de seguimiento
- Especificar y analizar los requisitos
- Diseñar e implementar la funcionalidad requerida
- Preparar y ejecutar las pruebas necesarias
- Actualizar la información de seguimiento e informes scrum
- Opcionalmente, realizar la auto-evaluación de lo realizado en el proyecto durante el sprint anterior, incluyendo las evidencias que argumentan cada una de las evaluaciones otorgadas.

Así pues, en cada sprint se debe haber completado una versión de todos los entregables, incluyendo documentos y software. Por otro lado, toda la información relativa a las tareas involucradas en el desarrollo deben quedar registradas y convenientemente actualizadas en la forja, para que tanto los alumnos como los profesores puedan observar el estado real del proyecto.

3.1.3. Cierre del proyecto

Durante la fase de cierre del proyecto, los integrantes del mismo deben completar los entregables documentales y el código fuente. Luego el profesor desplegará la aplicación (haciendo uso de las instrucciones proporcionadas en la documentación) sobre un servidor de aplicaciones propio o solicitará a alguno de los miembros del equipo a que lo lleve a cabo sobre algún entorno en la nube. Posteriormente realizará las pruebas (manuales y automáticas) pertinentes. Para la evaluación del proyecto, el profesor hará uso de una rúbrica de evaluación.

3.2. Roles

Los participantes asumirán una serie de roles típicos en cualquier proceso de desarrollo de software y en SCRUM, pudiendo una misma persona desempeñar varios roles, y un rol ser desempeñado por varias personas. Los roles posibles a desempeñar por los miembros de un equipo de proyecto son los siguientes:

- *Scrum Master*: hace cumplir las reglas y buenas prácticas, elimina impedimentos
- *Product Owner*: representante del cliente, prioriza historias de usuario
- *Arquitecto software*: diseña la arquitectura de los componentes y de todos los sistemas software que intervengan en el despliegue de la aplicación para ponerla en funcionamiento.
- *Analista*: analiza requisitos; realiza modelos de casos de uso y conceptuales, etc.
- *Diseñador de interfaz de usuario*: realiza mockups de pantallas y diseña y codifica la interfaz de usuario.
- *Programador*: diseña y codifica los componentes de software en Java.
- *Tester*: planifica, codifica y configura la ejecución de casos de prueba unitarios, funcionales y de integración.
- *Administrador de sistemas*: instala y configura los entornos de desarrollo, servidores y sistemas software de los que depende la aplicación (base de datos, servidor web, servidor de aplicaciones, sistema de control de versiones, etc.)

3.3. Herramientas

Herramientas de gestión

Para la gestión del proyecto se empleará una *forja* alojada en un espacio gratuito **privado** en *GitHub*.

El administrador de sistemas designado para el proyecto deberá crear un nuevo espacio **privado** de trabajo en la forja (Github). Este espacio deberá denominarse *iw2019-2020-ACRONIMO-DEL-PROYECTO* y se deberá otorgar permisos de

acceso (colaborador) al usuario *ruizrube*. Luego, en la página principal del proyecto se debe añadir información de bienvenida con los nombres de los miembros del equipo de trabajo y un enlace a la carpeta con los entregables. En Github se registrarán las historias de usuario y tareas como *issues* y los sprints como *milestones*. Se usarán las *releases* para colgar los ficheros empaquetados del software (ficheros jar).

Herramientas de comunicación

Además del correo electrónico, se recomienda hacer uso de alguna plataforma de mensajería para hacer más fluida la comunicación entre los miembros del equipo de trabajo.

Herramientas de diseño y modelado

A lo largo del proyecto será necesario usar herramientas para realizar diagramas de análisis, mockups, diseños, etc. Es potestativo de cada equipo de desarrollo el empleo de las herramientas con las que se encuentre más cómodo, aunque se requiere que sean de software libre o con licencia de uso no restringida para educación.

Herramienta de documentación

Google Drive será la plataforma que actuará como repositorio donde almacenar la documentación de los proyectos y las hojas de calculo para su evaluación.

El administrador de sistemas designado para el proyecto deberá crear una nueva carpeta privada en dicho sistema. Esta carpeta deberá denominarse *iw2019-2020-ACRONIMO-DEL-PROYECTO* y se deberá otorgar permisos de **escritura** al usuario *ivan.ruiz@gm.uca.es*. Esta carpeta deberá contener un fichero por cada uno de los entregables del proyecto. La documentación deberá estar en formato Google Docs (no siendo validos los formatos PDF u Word)-

Herramienta de desarrollo

Como entorno gráfico de desarrollo se aconseja usar una versión reciente de *Eclipse*, convenientemente configurado con los plugins necesarios. Se requiere el uso de *Apache Maven* como herramienta de gestión del proyecto, construcción, gestión de dependencias, documentación, pruebas y despliegue de la aplicación.

3.4. Entregables

La siguiente es una lista de los entregables (documentos) a generar. Se llevará a cabo un control de versiones de dichos entregables a través de las facilidades suministradas por el entorno colaborativo Google Drive. El código fuente se depositará, como ya se ha comentado, en la plataforma Github.

E1 *Documento de plan de proyecto*

E2 *Documento de análisis de requisitos*

E3 *Documento de diseño*

E4 *Documento de implementación*

E5 *Documento de pruebas*

E6 *Documento de implantación*

4. Entrega y evaluación

4.1. Hitos de entrega

Cada una de las iteraciones o *sprint* deben durar entre dos y tres semanas (según se planifique en las etapas iniciales del proyecto) y finalizará con la entrega de una versión del software y los documentos *entregables*, a depositar en la herramienta de edición colaborativa. Cada entrega parcial debe ser marcada con un hito (*milestone*) en la planificación que se haga del proyecto en la forja.

En la tabla 1 se presentan los hitos previstos de entrega. Los hitos 1 a N deberán fijarse por cada equipo en función de la estimación que se haga en la planificación inicial, y adecuarse conforme avance más o menos la ejecución del proyecto.

Hito	Descripción	Entrega
Hito 0	Entrega de Plan de proyecto	17 de octubre
Hito 1	<i>A definir por el equipo</i>	<i>A definir por el equipo</i>
Hito N	<i>A definir por el equipo</i>	<i>A definir por el equipo</i>
Hito Final	Entrega del proyecto y defensa obligatoria	16 de enero
Examen	Recuperación opcional	[Fecha conv. oficial de exámen]

Cuadro 1: Hitos del proyecto

4.2. Evaluación

La evaluación del proyecto se realizará atendiendo a una serie de criterios que vendrán recogidos en una *rúbrica* o instrumento de evaluación, disponible en el campus virtual. De este modo, se buscarán evidencias del cumplimiento de los criterios descritos anteriormente en la forja del proyecto y en el repositorio documental. Durante el proceso de evaluación, deberán estar presentes todos los miembros del equipo de trabajo, los cuales tendrán que responder a una serie de cuestiones sobre su implicación en el proyecto.

Es importante que haya un equilibrio en la distribución del trabajo, tanto en la asignación de roles, como en el reparto efectivo de tareas, el número de commits en el sistema de control de versiones y el volumen de ediciones sobre los documentos. De no ser así, las calificaciones serán ajustadas por exceso o por defecto a cada integrante del equipo.

La copia total o parcial de código fuente o de documentación procedente de proyectos realizados por otros compañeros será motivo suficiente para la no superación de la parte práctica de la asignatura. Se harán uso de herramientas de análisis para la detección automática de plagio.

El servidor donde se esté ejecutando el software deberá estar arrancado y preparado para el momento de la defensa. Al entrar en el aula se facilitará al profesor la URL de la aplicación para poder realizar las pruebas pertinentes. Se indicará la IP privada asignada al equipo, en el caso de usar un equipo portátil; o un nombre de dominio, en el caso de un servidor público. En el primer caso, comprobad que no haya ningún cortafuegos que impida acceder a la aplicación web. Asimismo, se deberá indicar las credenciales de acceso de un usuario con privilegios de administración.

La base de datos debe estar suficientemente poblada con información de los barcos, itinerarios, actividades, etc. para probar de manera satisfactoria las distintas características implementadas en la aplicación web.