

Enunciado Práctica

Diseña, implementa y despliega una aplicación web tolerante a fallos

NOTA: El contenido de este enunciado puede actualizarse y completarse a medida que avanza el curso. Cualquier actualización será notificada en los foros de la asignatura en el aula virtual.

Objetivo

Que el alumno implemente y despliegue una aplicación web tolerante a fallos de la forma más parecida a como lo haría a nivel profesional. Algunas de las características de la práctica que simulan un entorno real son las siguientes:

- La práctica se desarrollará en un equipo formado por 3 alumnos. De esa forma se simulará un equipo de trabajo real.
- La temática de la web será libre, de forma que los alumnos puedan usar su creatividad y su experiencia personal para definir las funcionalidades deseadas (dentro de unos límites que se definirán más adelante en el documento).
- Se utilizarán herramientas profesionales, tanto para el desarrollo en sí como para la compartición del código entre los miembros del equipo.
- Se automatizará la puesta en producción de la aplicación (despliegue)

Funcionalidad y características de la aplicación web

La aplicación web que hay que diseñar e implementar tiene una temática libre. El equipo de desarrollo puede decidir qué funcionalidades ofrece al usuario, el aspecto gráfico, el esquema de navegación, etc.

A modo de ejemplo, se presentan algunos tipos de aplicaciones web que se podrían implementar:

- Web de compra/venta de objetos usados
- Web de selección/contratación de cuidadores de niños por horas
- Web de un gimnasio con seguimiento de entrenamiento
- Web de una liga de fútbol
- Web de una guardería para consulta de actividades por los padres
- Web para diseñar actividades turísticas: lugares de interés, horarios, etc.
- Web de una academia de formación: cursos ofertados, activos, reserva, matrícula, etc.

- Web para docencia online: recursos, foros, alumnos, profesores, etc.
- Web de venta online: productos, carrito de la compra, stock, pedido, etc.
- Web de gestión de incidencias
- Web de reserva de aulas de informática en la universidad

Independientemente de la temática elegida por el equipo de desarrollo, la aplicación web deberá cumplir los siguientes requisitos técnicos:

- La aplicación web deberá implementarse con las tecnologías principales vistas en la asignatura: SpringBoot y MySQL. El despliegue se hará con Ansible.
- La aplicación web deberá disponer de una parte pública (accesible por los usuarios sin necesidad de hacer login) y una parte privada (a la que se accederá con usuario y password).
- El número de entidades que debe manejar la aplicación web será de al menos 5 y preferiblemente no más de 7. Estas entidades deberán estar relacionadas entre sí. Por ejemplo, si se opta por una tienda online, las entidades pueden ser, por ejemplo: producto, categoría de productos, pedido, usuario, oferta. En el caso de una web de búsqueda de cuidadores para niños, las entidades pueden ser: usuario, cuidador, comentario/valoración, disponibilidad y favoritos.
- La aplicación deberá estar dividida en un servidor web accesible públicamente y un servicio de uso exclusivo interno que se comunica con el servidor web a través de algunas de las técnicas estudiadas en el Tema 3. Este servicio interno será usado por el servicio web para tareas específicas, como por ejemplo:
 - Escalado de imágenes, si la aplicación soporta subida de imágenes por el usuario
 - Envío de correos a los usuarios
 - Generación de PDFs offline

Desarrollo de la aplicación web en fases

El desarrollo de la aplicación deberá realizarse siguiendo las siguientes fases:

- Fase 1: Formación del equipo de desarrollo, elección de la temática de la web, identificación de las entidades principales y definición de las funcionalidades del servicio interno.
- Fase 2: Aplicación web con base de datos en local.
- Fase 3: Aplicación web con base de datos MySQL en cloud.
- Fase 4: Aplicación web con balanceo de carga al menos para la parte web en cloud.
- Fase 5: Despliegue automatizado con Ansible

Estas fases deberán entregarse y presentarse al profesor en las siguientes fechas:

- Fase 1: 23 de Enero 2018
- Fase 2: 13/15 de Febrero 2018
- Fase 3: 13/15 de Marzo 2018

- Fase 4: 10/12 de Abril 2018
- Fase 5: 24/26 de Abril de 2018

La presentación de las fases se realizará de la siguiente forma:

- El contenido de cada fase se presentará al profesor en horario de clase.
- Aunque el trabajo es colaborativo y es importante el reparto de tareas, todos los miembros del equipo deberán conocer el código completo de la aplicación (aunque algunas partes hayan sido desarrolladas por otro compañero).
- Todos los miembros del grupo deberán estar presentes en la presentación porque el profesor podrá preguntar individualmente a cualquier alumno si lo considera necesario.
- Aunque el alumno tenga dispensa académica deberá presentarse a las presentaciones de las fases porque se consideran como una evaluación.

Proceso de desarrollo de la aplicación web

La aplicación web se desarrollará usando un repositorio de github¹. El código fuente de la aplicación tendrá licencia open source, en concreto, será Apache 2. El repositorio se creará en la cuenta de alguno de los miembros del equipo.

Para que el profesor tenga un registro de todos los grupos formados y los repositorios de GitHub usados por cada uno de ellos, los alumnos tendrán que darse de alta en el siguiente documento compartido de registro de equipos con fecha tope del **22 de Enero a las 16:00h**.

https://docs.google.com/spreadsheets/d/1zZZFoIKWQN7aGj6P4PYJS10IaPEg9OHEF0XDNB_anP0

Los miembros del equipo de desarrollo pueden coordinarse entre sí como consideren conveniente, aunque es recomendable que utilicen un tablero trello² o el servicio de github projects³ para gestionar el estado de las tareas y su responsable.

Durante el desarrollo de las fases 2, 3, 4 y 5 se irá subiendo el código al repositorio a medida que se vaya desarrollando. Es importante que los commits tengan un tamaño adecuado y no sean muy grandes. Es decir, no se considerará adecuado que una aplicación se implemente con un commit por cada fase. Es decir, el repositorio no sólo se utilizará para entregar la aplicación, si no que debe usarse para **desarrollar** la aplicación. Idealmente deberá realizarse un commit por cada funcionalidad que se añada a la aplicación.

Los mensajes de commit también tienen que ser adecuados y describir correctamente el objetivo del commit. Se recomienda que cada mensaje tenga en su primera línea un resumen del commit de no más de 50 caracteres y opcionalmente una descripción más extensa a partir de la tercera línea (dejando una línea en blanco entre el resumen y la descripción).

1 <https://github.com>

2 <https://trello.com>

3 <https://github.com/blog/2256-a-whole-new-github-universe-announcing-new-tools-forums-and-features>

Para gestionar el repositorio de github se puede usar cualquier cliente de git.

Cuando el código del repositorio esté listo para la entrega en alguna de las fases, será necesario hacer un tag en el repositorio con el nombre de la fase que se vaya a entregar. Por ejemplo, en la Fase 2, el tag se llamará “fase2”.

Antes de hacer el tag de cada fase, el fichero README.md deberá ampliarse para contener toda la información sobre la entrega que se va a realizar. No es necesario entregar una memoria en formato PDF.

El día concreto en el que se evaluará la práctica para un grupo se indicará en el documento de registro.

Fase 1: Equipo de desarrollo y temática de la web

En la fase 1 se debe formar el equipo de desarrollo, elección de la temática de la web, identificación de las entidades principales y selección de las funcionalidades del servicio interno.

Se creará un fichero README.md en la raíz del repositorio en el que se incluirá la siguiente información:

- Nombre de la aplicación web.
- Descripción de la temática de la web, indicando qué funcionalidad es pública y cual es privada y diferente para cada usuario.
- Nombre y descripción de cada una de las 4 o 5 entidades principales.
- Descripción de las funcionalidades del servicio interno
- Integrantes del equipo de desarrollo: Nombre, Apellidos, correo oficial de la universidad y cuenta en GitHub.

El fichero README.md se visualiza al acceder al repositorio desde un navegador web y debe estar escrito en formato Markdown para que se visualice correctamente.

Si se utiliza trello para la coordinación del equipo, deberá incluirse un link en el README que apunte al tablero.

Esta fase no es evaluable.

Fase 2: Desarrollo de la aplicación web en local

En la fase 2 se debe definir e implementar la aplicación web con base de datos.

Para realizar el diseño web, se utilizarán datos de ejemplo. Por ejemplo, si la aplicación web es una tienda virtual, para diseñar la página principal, se utilizarán productos de ejemplo. Es necesario planificar las pantallas que tendrá la aplicación y cómo se pasa de una pantalla a otra. Para generar el HTML se utilizará el motor de plantillas Mustache. Se decidirá qué páginas serán públicas y cuáles privadas, aunque de momento no será necesario implementar seguridad (todas las páginas serán públicas).

Para mostrar la navegación se creará un diagrama en el que se indicará desde qué página se puede navegar hasta otras páginas. Para ello, las páginas pueden ser capturas de pantalla de las páginas reales.

En esta fase debe diseñarse también el modelo de datos. Este modelo de datos debe incluir al menos cinco entidades. Al final de esta fase los datos deben persistirse en una base de datos MySQL, aunque durante el desarrollo se utilice una base de datos embebida (como H2).

En el **README** se deberán incluir **capturas de pantalla** de cada una de las páginas principales que hayan sido diseñadas y una **breve descripción de cada una** de ellas (un párrafo como mucho), así como el **diagrama de navegación**. También se debe incluir el **modelo de datos** de la aplicación utilizando un diagrama de clases **UML** y un diagrama **Entidad/Relación** que muestre cómo se persisten dichos datos en la base de datos relacional.

Fase 3: Inclusión de seguridad y servicio interno

En la fase 3 se debe implementar la aplicación web completamente funcional con las siguientes características:

- La aplicación gestionará los usuarios usando **Spring Security**. Se tendrá en cuenta que la seguridad está correctamente implementada, evitando que un usuario pueda acceder a una página sobre la que no tiene permisos y evitando que pueda realizar operaciones sobre elementos de la base de datos sobre los que no tiene permisos. Por ejemplo, si la aplicación es una red social, un usuario sólo podrá borrar sus propios mensajes, no los mensajes de otros usuarios.
- Implementar la **comunicación entre el servicio y la aplicación web** utilizando alguna de las técnicas del tema 3
- Definir la interfaz entre el servicio y la aplicación web: mensajes que se intercambian, el orden de los mismos, dónde se dejan los datos...
- Despliegue de la aplicación completa (web + servicio interno + bbdd) en una única máquina.

En el **README** se deberá incluir la siguiente información:

- **Navegación:** Se volverán a realizar las capturas de pantalla de las páginas principales usando la aplicación. En caso de que haya cambiado la navegación, se deberá actualizar el diagrama de navegación.
- **Diagrama de clases y templates:** Se creará un diagrama de clases de la aplicación. No se incluirán ni atributos ni métodos en las clases. Se mostrarán las relaciones entre las clases (asociación, composición o herencia) y se diferenciará claramente qué clases son **@Controller**, **@Service**, **Repository** u otro tipo de clases. Para ello se puede usar un código de colores, una distribución de las clases por partes u otro mecanismo. En este diagrama también se incluirán los templates usados y se indicará con qué **@Controller** se relacionan.
- **Instrucciones precisas para desplegar la aplicación:** compilación, cómo subir el jar, qué hace falta instalar en la máquina.

Fase 4: Incluir tolerancia a fallos en la aplicación

En esta fase no se introducirán funcionalidades adicionales. Esta fase está orientada a hacer la aplicación tolerante a fallos usando técnicas que permitan que la aplicación siga funcionando incluso cuando fallen algunos componentes o parte de la infraestructura.

En la fase 4 se deberán incluir técnicas del Tema 5 que permitan:

- Que la aplicación siga funcionando aunque se caiga algún nodo (servicio web, servicio interno o base de datos).
- Que un usuario logueado pueda acceder a sus páginas privadas con independencia de que haya un balanceador delante del servicio web.
- Que ciertas consultas se puedan cachear. Esto debe incluir la invalidación de la caché cuando sea preciso.

El README se deberá ampliar con:

- La documentación de la interfaz del servicio interno.
- La actualización del diagrama de clases y templates incluyendo las nuevas clases que pueda haber sido necesario incluir para implementar las características de tolerancia a fallos.
- Un diagrama que muestre la infraestructura en Azure y los diferentes componentes de la aplicación y cómo se comunican entre ellos.
- Un vídeo de unos 4 o 5 minutos indicando cómo usar la aplicación y explicando la infraestructura en cloud. El vídeo se subirá a YouTube y se incrustará al principio del README. Uno de los alumnos deberá grabarse la voz describiendo qué está haciendo según vaya interactuando con la aplicación web. El vídeo deberá mostrar los aspectos más relevantes de la aplicación web, entre otros:
 - Acceso de un usuario sin hacer login
 - Hacer login con un usuario existente y mostrar las funcionalidades adicionales
 - Si es necesario usar varios usuarios (usando varios navegadores web) para que se observe cómo las acciones de un usuario afectan a otro (mensajes, puntos, comentarios, solicitudes de amistad, etc.)
 - Creación de un nuevo usuario.
 - Modificación de los datos de una entidad de la aplicación web: libro, receta, coche, oferta, producto, etc. (la entidad concreta dependerá del tipo de web).
 - Comportamiento de la aplicación cuando se para un nodo del servicio web.
 - Comportamiento de la aplicación cuando se para un nodo del servicio interno.
 - Comportamiento de la aplicación cuando se para un nodo de la base de datos.

Fase 4: Incluir tolerancia a fallos en la aplicación

En esta fase se automatizará el despliegue de la aplicación. Concretamente, deben generarse los siguientes roles Ansible:

- Web
- ServicioInterno
- BaseDatos
- Balanceador

- Global: éste gobernará a todos los anteriores decidiendo desplegando todos los artefactos necesarios.

Evaluación

Al evaluar cada una de las fases se tendrán en cuenta los siguientes aspectos:

- La fase 1 no es evaluable. El resto de fases tendrán una nota entre 0 y 10.
- Durante la evaluación, el profesor puede detectar que la práctica tiene algunos aspectos mejorables. En ese caso, el profesor puede poner una nota condicionada a que esos aspectos sean subsanados en un plazo de 1 semana. Si no son subsanados, la nota final para esa fase sería inferior a la nota indicada.
- Si la fase no cumple con los mínimos exigidos (falta de funcionalidad, incompleta) o no funciona correctamente (tiene errores) se considerará suspensa. Un grupo puede recuperar una entrega suspensa si en una semana solventa las carencias graves que tiene. En una recuperación no se puede obtener más de un 8 (no se puede alcanzar la nota máxima de 10). Si en la primera recuperación se siguen detectando deficiencias, se podrá ofrecer una semana más adicional, pero en este caso no se podrá obtener una nota superior a 6. Si después de las 2 posibilidades de recuperación una fase queda suspensa, los alumnos no podrán realizar más entregas y tendrán que esperarse a la convocatoria extraordinaria para realizar las entregas que faltan y aprobar la asignatura.
- La nota de cada alumno del grupo es independiente. Un alumno puede obtener más o menos nota que el resto de sus compañeros. La nota individual de cada alumno se determina en base a:
 - El conocimiento de la aplicación durante la defensa, ya que todos los alumnos tienen que conocer el código de toda la aplicación aunque ellos no hayan desarrollado esa parte.
 - El histórico de commits de ese alumno en el repositorio. Los commits tienen que reflejar la aportación real del alumno a la práctica. Si tiene pocos commits o con poco contenido, se considerará que ha trabajado menos que el resto de compañeros.
- Los alumnos pueden realizar funcionalidad extra para obtener una nota superior a 10 en la práctica. Por cada funcionalidad extra implementada se podrá sumar hasta 0,5 puntos con un máximo de 12. Se considera funcionalidad extra la incorporación de alguna funcionalidad que no se haya visto en clase. Por ejemplo:
 - Utilización de una base de datos manejada en la nube (como Azure SQL server)
 - Uso de Redis como caché en lugar de la caché interna de Spring
 - Eliminar el punto único de fallo que supone el servicio interno introduciendo alguna técnica que permita tener varias instancias del servicio ejecutando para conseguir tolerancia a fallos en el servicio interno también
 - Uso de servicios externos como envío de correo
 - Otros (consultar con el profesor para ver si se considera funcionalidad extra).
- Si algunos miembros de un grupo deciden dejar la asignatura, el resto de miembros del grupo pueden continuar con el desarrollo de la misma aplicación web solos. También pueden

integrarse en otro grupo siempre que el número máximo de desarrolladores no supere 4 y que el grupo en el que se integren tengan aprobadas las mismas fases. Es decir, un alumno no se puede ir a un grupo que tenga aprobadas las fases 2 y 3 si él no tiene aprobada la fase 2 en su grupo original.

- Para obtener la nota final de la práctica se usará la nota de cada fase ponderada de la siguiente forma:
 - Fase 2: 25%
 - Fase 3: 25%
 - Fase 4: 35%
 - Fase 5: 15%
- Convocatoria extraordinaria:
 - Aquellas fases aprobadas en la convocatoria ordinaria se guardan para la convocatoria extraordinaria. Es decir, en la convocatoria extraordinaria los alumnos sólo tendrán que realizar aquellas fases suspensas o no entregadas.
 - La nota final de la práctica se calculará con los mismos porcentajes que en la convocatoria ordinaria.
 - En la convocatoria extraordinaria no existe una nota condicionada a que se solucionen los problemas encontrados.
 - En la convocatoria extraordinaria no existe la posibilidad de recuperar una fase suspensa. Si no se cumplen con las funcionalidades o la aplicación tiene fallos, la práctica (y por tanto la asignatura) se considerará suspensa.