



Grado en Ingeniería Informática

Sistemas Distribuidos

Parte I

©2023 Autor Nicolás H. Rodríguez Uribe

Algunos derechos reservados

Este documento se distribuye bajo la licencia

“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>



Contexto:

Esta asignatura tiene una única práctica, dividida en 3 partes bien diferenciadas, pero que van una a continuación de otra. El objetivo final, es tener una aplicación web, que conste:

- De una interfaz web (formularios) y una API REST. (Parte I)
- De información persistente, mediante una base de datos. (Parte II)
- De dockerización. (Parte III)

Es esencial tener en cuenta este objetivo final en cada una de las partes. En cada una de las partes se especificará el detalle de cada una de las mismas.

Temática:

La temática será la misma para toda la práctica. Esta temática será libremente escogida por los estudiantes. Por ejemplo, una aplicación web para una bodega/cervecería, torneos de pádel/tenis, restaurantes, etc.

Si bien la temática es libre, los requisitos mínimos serán comunes para todos los estudiantes.

Restricciones:

No será posible distinguir entre diferentes usuarios de una aplicación, así que toda la implementación será tanto de administrador como de usuario de la aplicación.

Grupos de estudiantes:

Los estudiantes realizarán la práctica en grupos de 2 o 3 personas. En casos excepcionales, se permitirá que se haga la práctica individual, previo visto bueno de los profesores.

En caso de separación, es necesario hablar con el profesor para estudiar la casuística.

Defensa de prácticas:

Los profesores podrán pedir defensa de alguna de las partes, o de todas. La defensa será parte de la nota de la práctica en caso de que se haga. Si no se acude a la defensa, la calificación será de un 0. Los miembros de un mismo equipo pueden tener distinta nota, según la defensa que hagan de la misma y su participación en el proyecto.

Repositorio:

Para un mejor seguimiento del trabajo, los estudiantes deberán invitar a los profesores a un repositorio privado de GitHub. El uso del mismo influye en la nota de la parte evaluada. A modo de ejemplo, una persona con 0 commits, tendrá de un 0.

Rúbrica:

Junto al enunciado de cada una de las partes de la práctica, se subirá la rúbrica que se utilizará para corregir la misma.

Tanto la rúbrica como el enunciado de la práctica pueden cambios menores a lo de la asignatura.

Enunciado:

En esta Parte I de la práctica se pide lo siguiente:

- Tres entidades con relación semántica entre ellas. **Relacionar las entidades entre si corresponde a la Parte II.**
- Toda información será volátil y no persistente. Se **recomienda** utilizar un mapa.
- Operaciones CRUD sobre cada una de las entidades a través de formularios. También es necesario implementar el equivalente a Patch a través de formularios.
- Operaciones CRUD sobre cada una de las entidades a través de la API REST. Es necesario también implementar la operación Patch.
- Una colección de Postman con las operaciones CRUD + Patch para cada una de las entidades.
- Para obtener la máxima calificación, también será necesario el uso de JavaScript y de CSS o similares.
- Se tendrá en cuenta el uso del repositorio.
- Los detalles de puntuación de cada una de las partes estarán reflejados en la rúbrica correspondiente a la misma.



Grado en Ingeniería Informática

Sistemas Distribuidos

Parte II

©2023 Autor Nicolás H. Rodríguez Uribe

Algunos derechos reservados

Este documento se distribuye bajo la licencia

“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>



Contexto:

Esta asignatura tiene una única práctica, dividida en 3 partes bien diferenciadas, pero que van una a continuación de otra. El objetivo final, es tener una aplicación web, que conste:

- De una interfaz web (formularios) y una API REST. (Parte I)
- De información persistente, mediante una base de datos. (Parte II)
- De dockerización. (Parte III)

Es esencial tener en cuenta este objetivo final en cada una de las partes. En cada una de las partes se especificará el detalle de cada una de las mismas.

Temática:

La temática será la misma para toda la práctica. Esta temática será libremente escogida por los estudiantes. Por ejemplo, una aplicación web para una bodega/cervecería, torneos de pádel/tenis, restaurantes, etc.

Si bien la temática es libre, los requisitos mínimos serán comunes para todos los estudiantes.

Restricciones:

No será posible distinguir entre diferentes usuarios de una aplicación, así que toda la implementación será tanto de administrador como de usuario de la aplicación.

Grupos de estudiantes:

Los estudiantes realizarán la práctica en grupos de 2 o 3 personas. En casos excepcionales, se permitirá que se haga la práctica individual, previo visto bueno de los profesores.

En caso de separación, es necesario hablar con el profesor para estudiar la casuística.

Defensa de prácticas:

Los profesores podrán pedir defensa de alguna de las partes, o de todas. La defensa será parte de la nota de la práctica en caso de que se haga. Si no se acude a la defensa, la calificación será de un 0. Los miembros de un mismo equipo pueden tener distinta nota, según la defensa que hagan de la misma y su participación en el proyecto.

Repositorio:

Para un mejor seguimiento del trabajo, los estudiantes deberán invitar a los profesores a un repositorio privado de GitHub. El uso del mismo influye en la nota de la parte evaluada. A modo de ejemplo, una persona con 0 commits, tendrá de un 0.

Rúbrica:

Junto al enunciado de cada una de las partes de la práctica, se subirá la rúbrica que se utilizará para corregir la misma.

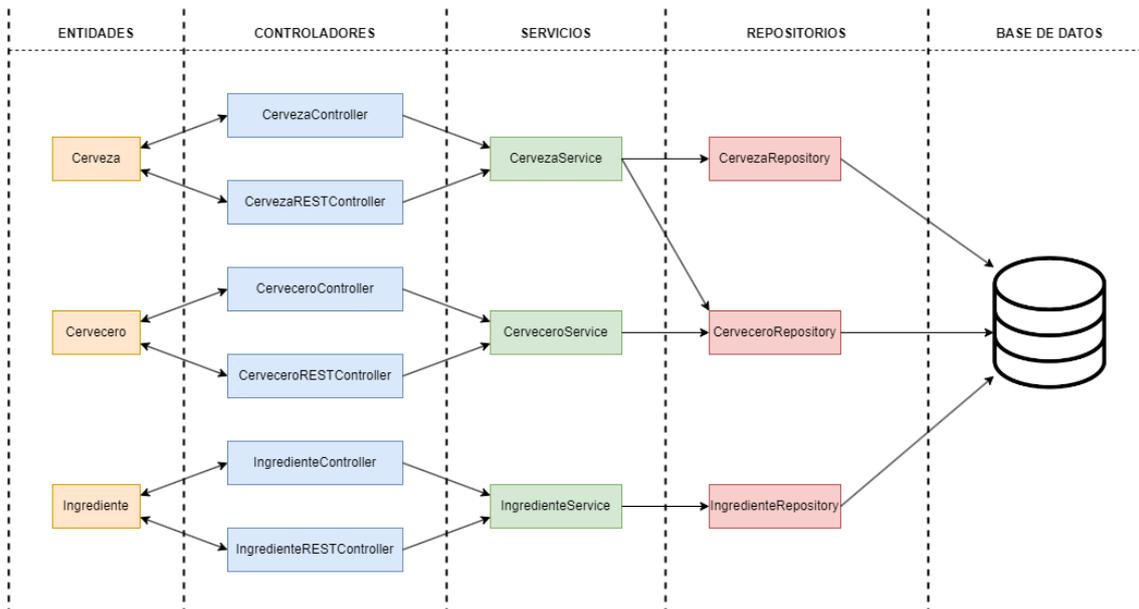
Tanto la rúbrica como el enunciado de la práctica pueden cambios menores a lo de la asignatura.

Enunciado:

En esta Parte II de la práctica se pide lo siguiente:

- Relacionar las entidades entre si, a través de los tipos de correspondencia 1:N y N:M.
- Toda información será persistente. Se **recomienda** utilizar H2, pero se puede utilizar MySQL.
- La conexión a la base de datos será mediante un @Service, y no a través de los @Controller ni @RestController.
- Los @Service serán utilizados por el @Controller y el @RestController asociados a una determinada entidad.
- Operaciones CRUD sobre cada una de las entidades a través de formularios. También es necesario implementar el equivalente a Patch a través de formularios. **Los cambios producidos por estas operaciones se verán reflejados en la base de datos.**
- Operaciones CRUD sobre cada una de las entidades a través de la API REST. Es necesario también implementar la operación Patch. **Los cambios producidos por estas operaciones se verán reflejados en la base de datos.**
- Una colección de Postman con las operaciones CRUD + Patch para cada una de las entidades.
- Los detalles de puntuación de cada una de las partes estarán reflejados en la rúbrica correspondiente a la misma.

Ejemplo:





Grado en Ingeniería Informática

Sistemas Distribuidos

Parte III

©2023 Autor Nicolás H. Rodríguez Uribe

Algunos derechos reservados

Este documento se distribuye bajo la licencia

“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>



Contexto:

Esta asignatura tiene una única práctica, dividida en 3 partes bien diferenciadas, pero que van una a continuación de otra. El objetivo final, es tener una aplicación web, que conste:

- De una interfaz web (formularios) y una API REST. (Parte I)
- De información persistente, mediante una base de datos. (Parte II)
- De dockerización. (Parte III)

Es esencial tener en cuenta este objetivo final en cada una de las partes. En cada una de las partes se especificará el detalle de cada una de las mismas.

Temática:

La temática será la misma para toda la práctica. Esta temática será libremente escogida por los estudiantes. Por ejemplo, una aplicación web para una bodega/cervecería, torneos de pádel/tenis, restaurantes, etc.

Si bien la temática es libre, los requisitos mínimos serán comunes para todos los estudiantes.

Restricciones:

No será posible distinguir entre diferentes usuarios de una aplicación, así que toda la implementación será tanto de administrador como de usuario de la aplicación.

Grupos de estudiantes:

Los estudiantes realizarán la práctica en grupos de 2 o 3 personas. En casos excepcionales, se permitirá que se haga la práctica individual, previo visto bueno de los profesores.

En caso de separación, es necesario hablar con el profesor para estudiar la casuística.

Defensa de prácticas:

Los profesores podrán pedir defensa de alguna de las partes, o de todas. La defensa será parte de la nota de la práctica en caso de que se haga. Si no se acude a la defensa, la calificación será de un 0. Los miembros de un mismo equipo pueden tener distinta nota, según la defensa que hagan de la misma y su participación en el proyecto.

Repositorio:

Para un mejor seguimiento del trabajo, los estudiantes deberán invitar a los profesores a un repositorio privado de GitHub. El uso del mismo influye en la nota de la parte evaluada. A modo de ejemplo, una persona con 0 commits, tendrá de un 0.

Rúbrica:

Junto al enunciado de cada una de las partes de la práctica, se subirá la rúbrica que se utilizará para corregir la misma.

Tanto la rúbrica como el enunciado de la práctica pueden cambios menores a lo de la asignatura.

Enunciado:

En esta Parte III de la práctica se pide lo siguiente:

- Añadir un base de datos relacional para la persistencia de datos.
- Utilizar contenedores de Docker para añadir la base de datos antes mencionada.
- Añadir una BBDD NoSQL o BBDD relacional distribuida.
- Los detalles de la evaluación están presentes en la rúbrica.



Grado en Ingeniería Informática

Sistemas Distribuidos

Parte I

1. Funcionalidad completa de la web en formularios
Completa | Faltan cosas menores (-2) | Faltan cosas importantes (-5)
2. Funcionalidad completa de la web en API REST
Completa | Faltan cosas menores (-2) | Faltan cosas importantes (-5)
3. Uso de GitHub
Completo | Poco uso (-1) | Solo una subida (-2) | Sin uso(-3)
4. Colección de Postman cargada con todas las operaciones de la API REST
Completo | Faltan algunas operaciones (-1) | No (-2)
5. Contiene estilos (CSS)
Completo | Elementos sin estilos (-0.5) | No (-2)
6. Contiene interactividad con el usuario (JavaScript)
Mucha | Media (-0.5) | No (-2)
7. Cumplimiento de las recomendaciones API REST: URL, status code, methods
Se cumple | Algunos fallos menores (-0.5) | Errores importantes (-2)
8. La aplicación tiene errores
No | Errores menores (-2) | Errores importantes (-3)
9. El código y los comentarios están en inglés
Si | No (-1)
10. El código tiene un formato y estilo adecuado.
Si | Algunas reglas no cumplen (-0.5) | No (-2)
11. Hay página de error
Si | No (-0.5)
12. Otras consideraciones (0-10)

©2023 Autor Nicolás H. Rodríguez Uribe

Algunos derechos reservados

Este documento se distribuye bajo la licencia

“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>





Grado en Ingeniería Informática

Sistemas Distribuidos

Parte II

1. Servicio aplicado al repositorio
Completo | Faltan alguna operación (-2) | Distintos (-3)
2. Servicios reutilizados entre los RestController y los Controller
Si | No (-2)
3. Cualquier tipo de cambio sobre un registro se reflejado en la BBDD (en H2)
Si | Tiene algún error (-2) | Faltan cosas importantes (-3)
4. Las entidades están relacionadas (1 : N)
Si | Tiene algún error (-2) | Faltan cosas importantes (-3)
5. Las entidades están relacionadas (N : M)
Si | Tiene algún error (-2) | Faltan cosas importantes (-3)
6. La aplicación tiene errores
No | Errores menores (-0.5) | Errores importantes (-1.5)
7. Otras consideraciones (0-10)

©2023 Autor Nicolás H. Rodríguez Uribe

Algunos derechos reservados

Este documento se distribuye bajo la licencia

“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>





Grado en Ingeniería Informática

Sistemas Distribuidos

Parte III

1. La aplicación tiene una base de datos.
 - Si, y es NoSQL o relacional distribuida
 - Si, pero es relacional no distribuida (-2.5)
 - No (-5)
2. La base de datos viene en un contenedor de Docker
Si | No (-5)
3. Otras consideraciones (0-10)



©2023 Autor Nicolás H. Rodríguez Uribe

Algunos derechos reservados

Este documento se distribuye bajo la licencia

“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>