# AUTOEVALUACIÓN DEL PROYECTO ShorterUrl

Objetivos

* Dominar nuevas tecnologías nuevas relacionadas con los Sistemas Web, para su posterior utilización, tales como: Node.js, Express, Angular, MongoDB (Full Stack Mean)
* Ser capaz de identificar las necesidades para desarrollar una aplicación Web que se despliega en varias máquinas

# Funcionalidades implementadas (60% de la nota del proyecto)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Peso | | Título | Nivel alcanzado |
| F1 | 25 | La aplicación permite busquedas de URI en el sistema. | | NOTABLE |
| F2 | 15 | La aplicación tiene cuentas de usuario | | SOBRESALIENTE |
| F3 | 20 | La aplicación devuelve mostrara estadísticas del sistema | | NOTABLE |
| F4 | **10** | La aplicación permite administrar permisos en las URL creadas | | SOBRESALIENTE |

# Justificación

| F1 | La aplicación permite búsquedas de URI en el sistema. |
| --- | --- |
| Notable | 1. Creación de un servicio Rest, que devuelva la n primeras URIs, que concuerden con los parámetros de búsqueda. 2. Se puede filtrar URI por los parámetros de creación de la URI 3. La aplicación permite asignarles información adicional en forma de descripción a las URI. 4. La aplicación permite buscar por Tags. |
| Verificación | 1. Se han creado pruebas automáticas: Test/UnitTest: 1º Test 2. Ejemplo en diapositvas: Ejecutar: <http://localhost:3000/findShort?description=noticias> 3. No hay test concreto, código en module/uri/controllers.js:createUrlShort() 4. Ejecutar http://localhost:3000/findShort?tags=peperoni |

| F2 | La aplicación permite búsquedas de URI en el sistema. |
| --- | --- |
| Sobresaliente | 1. Las URI creadas se asocian a un usuario anónimo. 2. Se podrán crear usuarios a través de un servicio Rest. 3. Cuando se crea una URI acortada se podrá asociar a la cuenta de un usuario concreto 4. Cuando un usuario se loguea recibe un token de sesión para poder realizar operaciones contra la API 5. Las URIs se podrán compartir entre usuarios. 6. Se podrán gestionar los links que se hayan creado 7. La sesión del usuario caducara transcurrido un tiempo |
| Verificación | 1. Se han creado pruebas automáticas: Test/UnitTest: 1º Test 2. Se han creado pruebas automáticas: Test/UnitTest: 2º Test 3. Se han creado pruebas automáticas: Test/UnitTest: 8º Test 4. Se han creado pruebas automáticas: Test/UnitTest: 6º Test 5. Se han creado pruebas automáticas: Test/UnitTest: 5º Test 6. No hay test concretos 7. No hay test concretos mismo sistema que en: Test/UnitTest: 7º Test |

| F3 | La aplicación devuelve mostrara estadísticas del sistema |
| --- | --- |
| Notable + | 1. Cada enlace acortado tendrá un recurso asociado con información de su creación que será la URI acortada seguida de un “info” 2. Existe un servicio que para un patrón de URI determinado devuelve información agregada 3. Devuelve estadísticas más complejas. |
| Verificación | 1. <http://localhost:3000/publicShort/pw52euyfv0c9dx6r/info> 2. <http://localhost:3000/publicShort/pw52euyfv0c9dx6r/info/click> 3. <http://localhost:3000/statistics/click> |

| F4 | La aplicación permite administrar permisos en las URL creadas |
| --- | --- |
| Sobresaliente | 1. Se podrán crear URI privadas que tendrán que una URI distinta a la pública. 2. Los usuarios podrán establecer una contraseña para acceder a sus URI privadas. 3. Los usuarios podrán establecer una contraseña para acceder a sus URI privadas. 4. Se podrán establecer una caducidad para el acceso de las URI privadas. |
| Verificación | 1. Se han creado pruebas automáticas: Test/UnitTest: 4º Test 2. Se han creado pruebas automáticas: Test/UnitTest: 5º Test 3. Se han creado pruebas automáticas: Test/UnitTest: 6º Test 4. Se han creado pruebas automáticas: Test/UnitTest: 7º Test |

# Tecnologías (20% de la nota del proyecto)

Rellenad la siguiente tabla indicando si se ha implementado (SI) o no (NO) la tecnología señalada en el proyecto.

|  |  |
| --- | --- |
| Tecnología | Implementada |
| RESTful cliente HTML/JavaScript y un servidor | SI |
| SOAP entre máquinas (no navegadores) | NO |
| RESTful entre máquinas (no navegadores) | NO |
| Websockets cliente HTML/JavaScript y un servidor | NO |

# Justificación

|  | RESTful cliente HTML/JavaScript y un servidor |
| --- | --- |
| Implementación | modules/routes.js |
| Verificación | * Todos los test atacan contra la API, Test/UnitTest: |

# Otras tecnologías (10% de la nota del proyecto)

|  | Promesas |
| --- | --- |
| Implementación | Librería “promise” en todos los services de los modules |
| Verificación | Aplicación montada sobre esta librería. |

|  | Lodash |
| --- | --- |
| Implementación | Librería para el procesado optimizado de arrays . Librería “lodash” en el service de statistics. |
| Verificación | Servicio montada sobre esta librería. |

|  | Mocha+Chai |
| --- | --- |
| Implementación | Librería para los test. |
| Verificación | Ejecutar los test, se verifican a si mismo. |