**PROPUESTA DE PROYECTOS DE DAM**

|  |  |
| --- | --- |
| **TÍTULO DEL PROYECTO** |  |
| **ALUMNO** |  |

Para aclarar la propuesta de proyecto y valorar su complejidad, podemos rellenar las siguientes tablas.

(Por favor, eliminad lo que no se use)

**FUNCIONALIDAD GENERAL**

La mayoría de aplicaciones necesitan una parte de **Frontend** y otra de **Backend**, pero se puede realizar una sola aplicación Frontend que lo realice todo.

Además se necesitará un **tipo de almacenamiento y acceso a datos** accesible desde las aplicaciones.

Es importante que las **tecnologías** a usar sean de las vistas en clase para que los profesores puedan dar soporte a las dudas que se planteen.

Si un alumno opta por alguna tecnología no estudiada en ningún módulo, no dispondrá de ayuda del profesorado por lo que puede quedarse bloqueado y no terminar el proyecto.

No es obligatorio utilizar todas las tecnologías estudiadas, pero sí algunas que representen lo aprendido durante el ciclo.

De las tecnologías usadas, en alguna de las partes debe utilizarse alguna que no sea WEB, es decir, aplicación de **móvil** o **escritorio**.

|  |  |
| --- | --- |
| **FRONTEND (IDE y Lenguaje ):** |  |
| **OPCIÓN DE MENÚ** | **DESCRIPCIÓN** |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **BACKEND (IDE y Lenguaje ):** |  |
| **OPCIÓN DE MENÚ** | **DESCRIPCIÓN** |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **TABLAS EN CASO DE USAR BASES DE DATOS SQL** | |
| **TABLA** | **DESCRIPCIÓN** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**FUNCIONALIDAD EXTRA QUE AUMENTA COMPLEJIDAD**

La incorporación de complejidad extra en la propuesta de proyecto supone que la funcionalidad general pueda ser más reducida.

|  |  |  |  |
| --- | --- | --- | --- |
| **FUNCIONALIDAD** | **POSIBLES OPCIONES** | **DESCRIPCIÓN** | **OPCIONES USADAS** |
| Identificación de usuarios  (en Frontend,  Backend  y/o APIREST) | Validación  Registro  CRUD  Validación de email  Uso de cookies, sesiones, JWT, Oauth, ...  ... | Las aplicaciones pueden ser de acceso público o con identificación.  En caso de necesitar que el usuario se identifique para mostrar funcionalidades distintas, podremos incorporar algunos aspectos de identificación |  |
| APIREST con SWAGGER | Swagger con JSON  Swagger con anotaciones | Para poder mostrar los métodos ofrecidos en el APIREST podemos usar asistentes como Swagger. |  |
| Imágenes | Archivo local  Archivo en FTP/SFTP  Archivo en Blob Storage  Imagen en BD con Base64 | El almacenamiento de imágenes puede realizarse en local como archivo, en remoto con FTP/SFTP, en la nube con algún Blob Storage e incluso en la base de datos codificando en Base64. |  |
| PDF |  | La generación de archivos PDF permite generar informes que pueden tratarse mediante visores en cualquier plataforma. |  |
| Códigos QR |  | La generación de códigos QR nos permite el acceso a información y URLs de forma directa al leerlos con la cámara del móvil, para luego realizar alguna acción. |  |
| Gráficos | Barras, circulares, mapas, líneas, ... | Mostrar gráficos estadísticos en pantalla permite crear interfaz con resultados más comprensibles y aparentes. |  |
| Animaciones |  | La posibilidad de mostrar algún tipo de animación nos permite crear información atractiva. |  |
| Sensores |  | En las aplicaciones móviles podemos utilizar alguno de los sensores disponibles y por lo tanto crear Apps muy útiles en diferentes entornos. |  |
| ... |  |  |  |

**ACCESO A FICHEROS**

Si se usan archivos en la aplicación (como datos temporales, de configuración, fotos o imágenes, ...) que deben ser almacenados en algún dispositivo:

|  |  |  |  |
| --- | --- | --- | --- |
| **TIPO DE ACCESO** | **LUGAR** | **TECNOLOGÍA** | **USADO** |
| Ficheros | Local | Almacenamiento en dispositivo local |  |
| Ficheros | LAN o Cloud | Carpeta compartida, FTP/SFTP, ... |  |

**ACCESO A DATOS DE BASE DE DATOS**

Para almacenar los datos podemos usar diferentes tipos de estructuras de acceso a bases de datos.

En el proyecto se pueden usar uno o varios tipos de acceso a datos en las aplicaciones.

Por ejemplo:

|  |  |  |  |
| --- | --- | --- | --- |
| **TIPO DE ACCESO** | **LUGAR** | **TECNOLOGÍA** | **USADO** |
| JDBC | Local | SQLite, H2, ... |  |
| JDBC para BD SQL | LAN o Cloud | MySQL, Oracle, SQL Server, … |  |
| JDBC para noSQL | LAN o Cloud | MongoDB, redis, Firebase, … |  |
| APIREST DAO - Local | Servidor en LAN | (Apache Tomcat o Wildfly) + BD |  |
| APIREST JPA - Local | Servidor en LAN | (Apache Tomcat o Wildfly) + BD |  |
| APIREST PHP – Local | Servidor en LAN | (XAMPP/LAMP) + BD |  |
| APIREST – Local | Servidor en LAN | Otras tecnologías como Node.js Express, Flask, … (sin soporte de profesores) |  |
| APIREST DAO - Cloud | Azure o AWS | (Ubuntu) + (Apache Tomcat o Wildfly) + BD |  |
| APIREST JPA - Cloud | Azure o AWS | (Ubuntu) + (Apache Tomcat o Wildfly) + BD |  |
| APIREST - Cloud | Servidor en Cloud | Otras tecnologías como Node.js Express, Flask, … (sin soporte de profesores) |  |
| APIREST - Open Data | Servidor en Internet | Acceso a APIREST de terceros existente en Internet con o sin API\_KEY |  |
|  |  |  |  |
|  |  |  |  |