

# Requerimientos del proyecto final

*Tecnología de componentes.*

# Requerimientos del proyecto final

*Tecnología de componentes*

## Introducción.

El objetivo de este proyecto es realizar dos pequeños softwares que permitan evaluar los conocimientos aprendidos por el alumno durante el curso de **tecnología de componentes** y fomentar buenas prácticas en el desarrollo de productos de software bajo las tecnologías, estándares y los paradigmas explorados en clase.

La presentación y aprobación de este proyecto es indispensable para la acreditación la asignatura.

Las características a cumplir con la totalidad o parcialidad del proyecto serán expresadas en este documento.

## Características generales del proyecto.

### Descripción

El software consiste un mapa que permita registrar y consultar la información sobre lugares. El desarrollo del proyecto está dividido en 2 piezas de softwares distintos:

- Backend: Una API REST
- Frontend: Software de usuario final

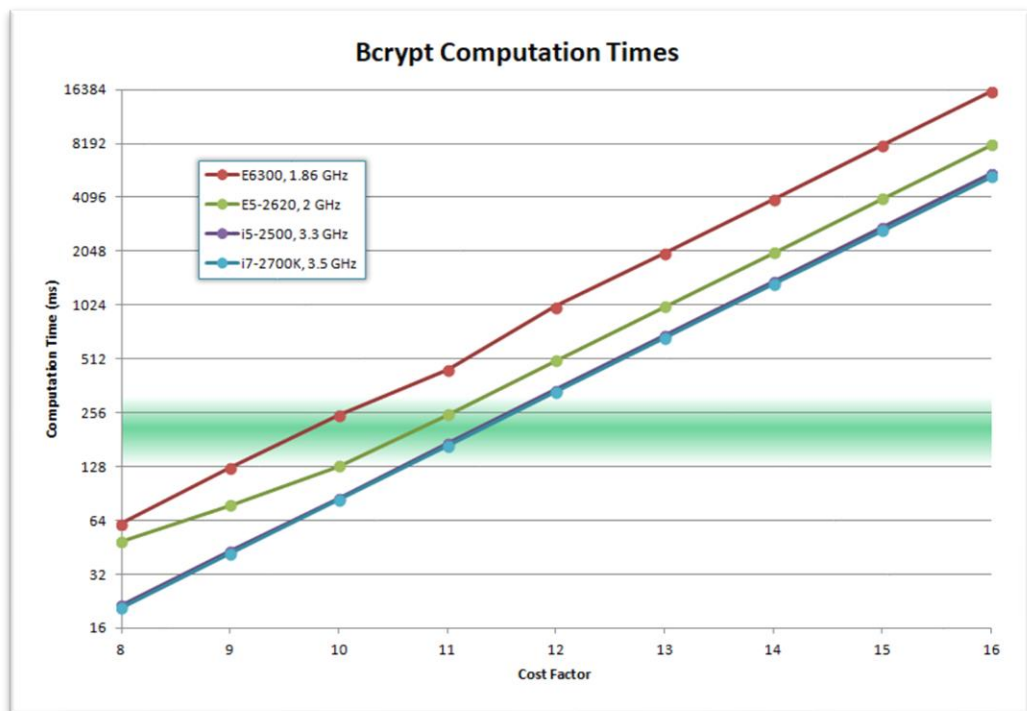
### Requerimientos funcionales (Historias de usuario).

- Como usuario quiero poder acceder al mapa.
- Como usuario quiero ver los lugares marcados en el mapa.
- Como usuario quiero ver mi ubicación actual.
- Como usuario quiero ver los detalles de un lugar del mapa al seleccionarlo.
- Como usuario quiero ver una foto por lugar.
- Como usuario quiero poder hacer una búsqueda de lugares por nombre.
- Como usuario quiero poder ver las calificaciones de un lugar. **[no obligatorio]**
- Como usuario no autenticado quiero poder registrarme en plataforma.
  - Con mis datos personales y correo
  - Con al menos una red social o servicios de Google **[extraordinario]**
- Como usuario no autenticado quiero poder iniciar sesión a una cuenta.
- Como usuario autenticado quiero poder cerrar sesión.
- Como usuario autenticado quiero poder registrar lugares.
- Como usuario autenticado quiero poder calificar un lugar. **[no obligatorio]**

- Como usuario autenticado quiero dar de baja mis lugares.
- Como usuario autenticado quiero editar mis lugares.

#### Requerimientos no funcionales.

- **Requisitos del producto:**
  - Las contraseñas del usuario deben ser encriptados con bcrypt.
  - El coste de factor de encriptación no debe ser mayor a 12 ni menor a 9. Decida dependiendo del procesador que está utilizando.



- La autenticación del usuario debe ser validada a través de JSON Web Tokens.
- La API REST debe poder ser consumido por orígenes desconocidos (evite errores por CORS).
- El sistema de gestión de base de datos debe ser MySQL o alguna de sus bifurcaciones.
- El manejo de la base de datos, así como el acceso a los modelos debe ser a través de un ORM.
- Se recomienda el uso de un estado centralizado en el software cliente según sea necesario para evitar el uso de variables globales.
- El JWT debe ser almacenado en frontend.
  - Almacenarlo en cookies [extraordinario]

### Requisitos organizativos:

- La construcción del software del usuario final y los servicios deben ser desarrollado bajo el paradigma de componentes.
- Uso del patrón de diseño **atomic design** para el desarrollo del software cliente.
- Tanto el backend como el frontend deben obedecer una misma guía de estilo.
- Ambas piezas de software deben estar en un repositorio público.
  - Ambos repositorios deben estar licenciados bajo MIT.
  - Se recomienda el uso de projects para gestionar la organización de las historias de usuario y los requisitos del producto.
  - Los repositorios deben ser público.
  - Los repositorios deben de agregar como colaboradores a los desarrolladores y a la cuenta [mpargar](#).

### Condiciones de entrega y evaluación.

- La fecha límite para el último commit es el 30 de mayo.
- La revisión ~~del proyecto~~ será el 31 de mayo.
- Periódicamente sus códigos serán revisados y se les avisará de posibles errores en la sección de issues.
- La entrega puede ser individual o en parejas.
- Se deben cumplir con todos los puntos obligatorios (aquellos no marcados como no obligatorios o extraordinarios) para la acreditación total (100%) del proyecto.
- Es posible realizar la actividad con equipos de 3 integrantes, pero estos deberán cumplir con los las características **no obligatorias**.
- Es posible realizar la actividad con equipos de 4 integrantes, pero estos deberán cumplir con los las características **no obligatorias** y **extraordinarias**.
- Aquellos alumnos que hayan fallado alguna unidad, trabajos o necesiten subir la calificación podrán realizar los puntos no obligatorios o extraordinarios, consulte con el profesor para aclarar detalles **[solo aplica a equipos de 2 o 3 integrantes]**
- Todos los colaboradores deben realizar contribuciones significativas en al menos 1 de ambos repositorios.
- Se pueden perder puntos en la evaluación (individual o grupal dependiendo el caso) si:
  - No se cumplen con las características del del proyecto o son deficientes.

- El diseño y construcción del software es deficiente.
- El software no es funcional.
- El software contiene errores que afectan la experiencia de usuario.
- Las contribuciones de un colaborador son nulas o insuficientes.