

## **Requisitos Funcionales**

### GENERAL:

RF1. Log in

### DUEÑOS DE NEGOCIOS:

RF2. Gestionar negocio

RF3. Realizar ofertas

### PEREGRINOS:

RF4. Gestionar rutas y sitios de interés

RF5. Valorar rutas y sitios de interés

RF6. Consultar descuentos

### DUEÑOS DE PILGRIMFY:

RF7. Consultar estadísticas

RF8. Promocionar negocios

## **Casos de Uso**

Debido a la implementación de un programa cliente servidor, realizaremos 3 aplicaciones, una por cada uno de los clientes y además una aplicación para el servidor al que se conectarán los mismos.

Hemos mapeado los requisitos a los siguientes casos de uso. Cada uno tiene una prioridad. La prioridad se ha establecido principalmente por la necesidad de tener funcionalidades para poder implementar otras que dependan de ellos. Aunque también dependen del grado de importancia que establezcamos que tenga ese caso de uso con respecto al uso y funcionamiento principal de la aplicación.

## **Prioridad**

**RF.2. Gestionar negocio.** Es el más prioritario ya que necesitamos de sitios de interés, entre los cuales están los negocios que subirán a la aplicación los dueños y a partir de la cual quedará definida la línea base de la arquitectura. No depende de ningún requisito.

**RF4. Gestionar rutas y sitios de interés.** Los peregrinos podrán planificar rutas, consultar las rutas y los sitios de interés oficiales, ofreciéndoles un servicio para filtrado por localización. También podrán crear un cuaderno de bitácora y compartirlo. Depende de RF2.

**RF3. Realizar ofertas.** Los dueños de los negocios podrán realizar ofertas a los peregrinos. Depende de: RF2

**RF5. Valorar rutas y sitios de interés.** Los peregrinos podrán realizar valoraciones y comentarios sobre las rutas y los sitios de interés. Depende de: RF2 y RF4.

**RF1. Log in.** Común a todos los clientes. No depende de ningún requisito.

**RF6. Consultar descuentos.** Los peregrinos podrán consultar los descuentos que les ofrecen los dueños de negocios. Depende de RF3.

**RF7. Consultar estadísticas.** Los dueños de Pilgrimfy podrán consultar las estadísticas de la aplicación, como puede ser el número de negocios oficiales, el número de usuarios registrados y activos en la aplicación, valoraciones sobre el servicio. Depende de RF2 y RF5.

**RF8. Promocionar negocios.** Los dueños de Pilgrimfy podrán realizar promociones a los negocios que consideren, ya sea por buenas valoraciones, más visitados...Depende de: RF7.

CdU	Requisitos	Iteración	Prioridad	Nombre	Versión
1	RF.1	It4	4	Log in	
2	RF.2	It1	1	Gestionar negocio	
3	RF.3	It3	3	Realizar ofertas	
4	RF.4	It2	2	Gestionar rutas y sitios de interés	
5	RF.5	It3	3	Valorar rutas y sitios de interés	
6	RF.6	It4	4	Consultar descuentos	
7	RF.7	It3	3	Consultar estadísticas	
8	RF.8	It4	4	Promocionar negocios	

## Estimación del tiempo

Hemos realizado la siguiente estimación de tiempo. Se mostrarán varias tablas, una por cada aplicación y una para el servidor. Además, tendremos una tabla con el tiempo total agrupado por cada caso de uso, que será la que utilicemos para crear nuestra división de trabajo y el calendario.

### **(MIRAR EL EXCEL, La página de estimación de tiempo)**

En total tendremos un proyecto con una duración de 409 horas. Solaparemos las iteraciones con el mismo nivel de prioridad intentando minimizar el número de días para la entrega del proyecto y además intentaremos minimizar la fragmentación interna en el tiempo.

## **GESTIÓN DE LA CONFIGURACIÓN**

- **Gestión de Versiones**
- **Gestión de Cambios**
- **Proceso de Almacenamiento de Incrementos y Versionado**
- **Ramificación**
- **Proceso de Construcción del Sistema**
- **Proceso de Liberación**

## **GESTIÓN DE VERSIONES**

### **Versionado de las Aplicaciones**

Cada aplicación llevará un versionado que dependerá de los requisitos implementados en la misma. El versionado aparecerá representado por X.Y.Z

- X: mayor: cambios que rompen con la compatibilidad
- Y: minor: cambios que amplían las funcionalidades de la aplicación
- Z: patch: cambios para arreglar fallos

### **Dueños de negocios**

V.1.1.0 → Realización del RF.2

V.1.2.0 → Realización del RF.3

V.1.3.0 → Realización del RF.1

### **Peregrinos**

V.1.1.0 → Realización del RF.4

V.1.2.0 → Realización del RF.5

V.1.3.0 → Realización del RF.1 y RF.6

### **Dueños de Pilgrimfy**

V.1.1.0 → Realización del RF.7

V.1.2.0 → Realización del RF.1 y RF.8

### **Servidor**

V.1.1.0 → Realización del RF.2

V.1.2.0 → Realización del RF.4

V.1.3.0 → Realización del RF.3, RF.5 y RF.7

V.1.4.0 → Realización del RF.1, RF.6 y RF.8

Además de las aplicaciones, los archivos y documentación que generemos llevará un versionado:

**Archivos de Visual Paradigm:**

Cada archivo tendrá un nombre con el formato de la versión que esté implementando: <NombreAplicacion>\_X.Y.Z.vpp

**Archivo Excel:**

Cada archivo Excel tendrá un nombre con el formato:  
<Planificación>\_X.Y.Z.xlsx

V.1.1.0 → Realización de la estimación del tiempo

V.1.2.0 → Realización de la división del trabajo y del calendario de trabajo

**Documentación:**

Cada archivo de la documentación tendrá un nombre con el formato:  
Documentación <NombreAplicacion>\_X.Y.Z.xlsx

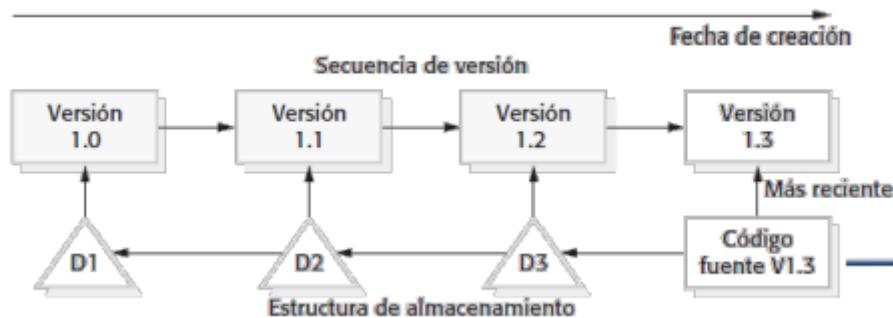
Los valores X.Y.Z corresponderán con la versión documentada de las funcionalidades de la aplicación hasta el momento.

## **GESTIÓN DE CAMBIOS**

En el caso de que tengamos que realizar algún cambio, se creará un issue que tendrá el formato: "PC <NombreAplicación>\_X.Y.Z" indicando para ese archivo o aplicación cual será el cambio que se quiere realizar. La directora del trabajo valorará si es conveniente realizar dicho cambio analizando los costes y beneficios. En el mensaje, se deberá indicar el nuevo nombre que tendrá el artefacto.

## PROCESO DE ALMACENAMIENTO DE INCREMENTOS Y VERSIONADO

Para el almacenamiento y el control de versiones de las aplicaciones utilizaremos un repositorio de github. Cuanto mayor sea el numero de la versión, más reciente será. Siempre que queramos incrementar la funcionalidad de un componente de la aplicación, deberemos utilizar la ultima versión almacenada en el repositorio.



## RAMIFICACIÓN

Se crearán las siguientes ramas:

- **Ramas de desarrollo**
  - Rama principal (**develop**) con la revisión de la aplicación durante el desarrollo o mantenimiento. Se corregirán los defectos no urgentes.
  - Una rama por cada iteración que afecte a un componente. Si tenemos varios componentes se crearán tantas ramas como afecten a dichos componentes en la iteración. Se creará con el formato "itx: <NombreAplicacion>"
- **Ramas de release**
  - Una rama para cada fase del plan de liberalización. Se creará con el formato "Release <NombreFase>"
    - Release Alfa
    - Release Beta
    - Release Final
- Una rama para los artefactos: archivos de Excel, Documentación sobre el uso del programa.
- Rama Master: con el código listo para desplegar en el entorno de producción
- Rama Hotfix: Corrección de los defectos graves encontrados en producción

## PROCESO DE CONSTRUCCIÓN DEL SISTEMA

Utilizaremos Maven para la construcción del sistema ejecutable al completo a partir de los distintos componentes generados. Gracias a Maven, podremos integrar estos componentes con la mínima recompilación.

## PLAN DE LIBERALIZACIÓN

El proyecto tendrá un plan de liberalización que constará de 3 fases:

- **Versión Alfa:** Publicada tras alcanzar el producto mínimo vital tras acabar la fase de construcción, el día x/x.
- **Versión Beta:** Publicada tras analizar y corregir los errores de la versión beta durante la fase de transición, el día x/x. Los usuarios nos podrán seguir dando feedback sobre errores.
- **Versión Final:** Publicada al acabar la fase de transición donde todos los errores tomados hasta el momento serán arreglados. Lanzada el día x/x