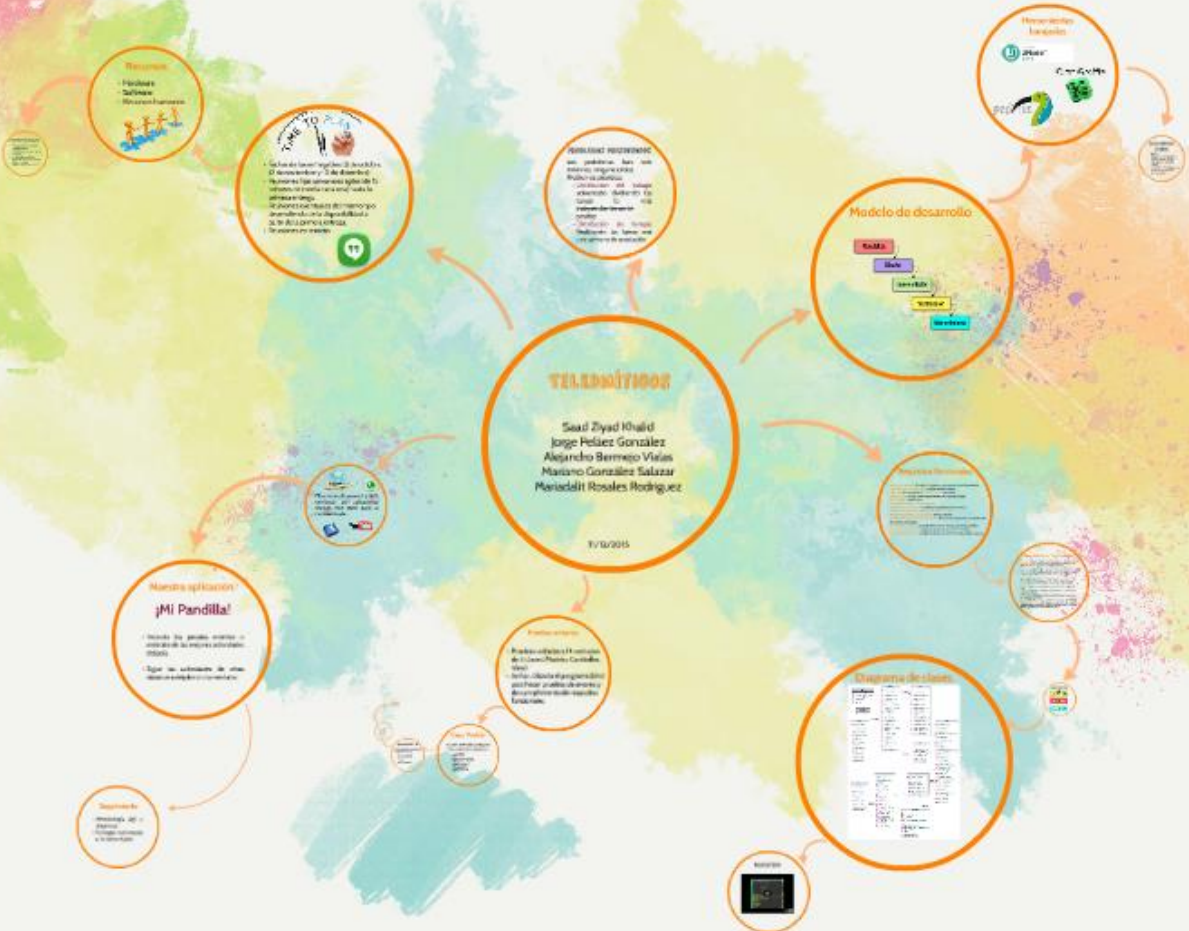


SOFTWARE DE COMUNICACIONES



TELEDMÍTICOS

Saad Ziyad Khalid
Jorge Peláez González
Alejandro Bermejo Vialas
Mariano González Salazar
Mariadalit Rosales Rodríguez

11/12/2015



El aumento de usuarios y de la demanda de aplicaciones sociales han dado paso al nacimiento de....





Nuestra aplicación:

¡Mi Pandilla!

- Anuncia tus propios eventos o entérate de las mejores actividades del país.
- Sigue las actividades de otros usuarios o déjales un comentario.

Seguimiento

- Metodología ágil y dinámica
- Entregas numerosas e incrementales



- Fechas de los entregables (8 de octubre, 12 de noviembre y 10 de diciembre)
- Reuniones fijas semanales ágiles (de 15 minutos de media cada una) hasta la primera entrega.
- Reuniones eventuales del mismo tipo dependiendo de la disponibilidad a partir de la primera entrega.
- Reuniones en remoto.



Recursos

- Hardware
- Software
- Recursos humanos



PAQUETES DE TRABAJOS

- 1.Aprobación del proyecto
- 2.Organización
- 3.Realización de documentos
- 4.Diseño
- 5.Desarrollo del código
- 6.Fase de pruebas de errores
- 7.Administración
- 8.Presentación del sistema

Modelo de desarrollo



Herramientas barajadas



OmniGraffle



HERRAMIENTAS USADAS

- Java 1.9
- Dropbox como repositorio
- “DDBB” ficheros de texto exportables/importables
- Umodel
- Foro UC3m / Whatsapp como medio de comunicación
- Librerías de Java

Requisitos funcionales

- Registro de usuarios: Permite el registro y almacenamiento de usuarios.
- Tipos de usuarios registrados: normal y patrocinador.
- Seguridad en contraseñas e identificación de cuenta.
- Accesibilidad: visualización dependiendo del tipo de usuario.
- Perfil y gestión del mismo.
- Visualización de eventos y usuarios.
- Seguimiento de usuarios: posible el seguimiento de usuarios.
- Creación, modificación y eliminación de eventos.
- Búsqueda de eventos e inscripción en los mismos.
- Modificación y eliminación de cuentas de usuario: se permite la modificación de cuentas de usuario.
- Búsqueda de usuarios: aplicado de la misma forma que en los eventos.
- Comentario en eventos: posible la acción de comentar eventos.
- Comentario en perfiles: posible la acción de comentar tableros de usuario.

Requisitos no funcionales

- **Usabilidad:** fácil de usar con respecto al usuario y por todo tipo de personas.
- **Confiabilidad:** existencia de medidas de seguridad como requerir una contraseña de usuario y almacenaje de datos duplicable.
- **Rendimiento:** interacción rápida y eficiente (tiempo de procesamiento en el que el usuario tiene la aplicación bloqueada, no excede de 0.5 segundos).
- **Compatibilidad:** no necesarias extensiones para funcionar. Mantenimiento por los empleados de mantenimiento de la empresa contratante.
- **Implementación:** no existen restricciones en la plataforma de hardware a utilizar ni por parte del grupo de mantenimiento o de testing.
- **Interfaz:** la interfaz de la aplicación será una terminal de comandos.
- **Claridad:** La representación de la información clara y diferenciada.
- **Operación:** gestión será realizada por el equipo de implementación del mismo.
- **Despliegue:** instalación realizada por el equipo de mantenimiento de la empresa contratante.
- **Mantenible:** con facilidades a la hora de realizar pruebas, cambiar cosas o realizar análisis del sistema.
- **Legal:** el sistema ha de ser legal, con una licencia de Creative Commons.
- **Responsabilidades de fallo:** asumidas por el equipo de implementación si el fallo corresponde a un proceso, o de la empresa contratante si el fallo es debido a algún problema de hardware o personal.

Pruebas unitarias

- Pruebas unitarias a 14 métodos de 3 clases (Modelo, Controller, View)
- Se ha utilizado el programa JUnit para hacer pruebas de errores y de cumplimiento de requisitos funcionales

Clase Modelo

Cuatro métodos probados
con resultados correctos:

- `getID()`
- `getNombre()`
- `getLugar()`
- `getFecha`

CLASE CONTROLLER

Cuatro métodos probados con resultados correctos:

- buscarUsuarios()
- limpiarConsola()
- inicio()
- menuPrincipal()

Clase View

Seis métodos fueron probados, obteniendo resultados correctos:

- mostrarPerfil()
- mostrarUsuarios()
- mostrarEventos()
- limpiarPantalla()
- mostrarInicio()
- mostrarLogin()

Lo que se ha probado

Requisitos funcionales en el ERS .

- Ejemplo: la clave cumple unas ciertas características y sino muestra un mensaje de error
- Ejemplo 2: cuando un usuario se registra, su perfil se guarda correctamente en un fichero para almacenar los usuarios

PROBLEMAS PRESENTADOS

Los problemas han sido mínimos, ninguno crítico.

Problemas previstos:

- **Distribución del trabajo:**
solventado dividiendo las tareas lo más independientemente posible
- **Distribución de tiempo:**
finalizando las tareas con una semana de antelación.