



universidad
de león



Ayuntamiento de Carrocera

Ingeniería del Software II

Alumnos:

Alberto Gutiérrez Morán

Miguel González García

Miguel Ángel Pérez López



Índice

Descripción de la funcionalidad final de la aplicación	3
SCRUM	5
Sprint 1	5
Descripción	5
Contenido de la pila	5
Acta de reunión de planificación	7
Acta de reunión de revisión	8
Acta de reunión de retrospectiva	9
Sprint 2	10
Descripción	10
Contenido de la pila	11
Acta de reunión de planificación	12
Acta de reunión de revisión	13
Acta de reunión de retrospectiva	14
Pruebas de unidad	15
Prueba de caja blanca	15
Cálculo de la complejidad ciclomática	15
Prueba de caja negra	18
Partición de equivalencia	18
Identificación de las clases de equivalencia	18
Creación de los casos de prueba	19
Controlador de versiones	20
Ejecución	22



Descripción de la funcionalidad final de la aplicación

La aplicación web realizada es una web ficticia del Ayuntamiento de Carrocera, en la provincia de León. Cuenta con funciones que no existen en el ayuntamiento realmente, por ejemplo, reserva de pistas de tenis o pádel, que no hay. Pero añadir éstas funcionalidades nos permitió hacer una aplicación más completa y que concuerde con los requisitos.

Las funcionalidades finales de la aplicación concuerdan con las descritas en el documento de la memoria descriptiva ya entregado en el mes de marzo. La descripción de las funcionalidades divididas por grupos son:

- VISUALES
 - Información general del ayuntamiento.
 - Información sobre todos los pueblos pertenecientes al ayuntamiento.
 - Datos y estadísticas demográficas sobre el municipio.
 - Información sobre turismo, ocio, fiestas o espacios naturales
 - Fecha y localización sobre eventos.
 - Noticias del ayuntamiento.
- FUNCIONALIDADES GENERALES
 - Existencia de tres roles: administrador, usuario empadronado y usuario normal.
 - Inicio de sesión en la página web: como administrador o usuario.
 - Registrarse en la página web: solo es posible registrarse como usuario normal o empadronado. Para éste último se verifica que el DNI con el que se registra el usuario está en el padrón municipal.
- FUNCIONALIDADES USUARIO NORMAL
 - Modificar los datos personales del usuario.



- Visualizar los eventos y apuntarse a los que tengan inscripción.
- Reserva de actividades ofrecidas por el ayuntamiento.
- Cerrar sesión.
- **FUNCIONALIDADES EMPADRONADO**
 - Visualizar libros de las bibliotecas y reservarlos o devolverlos.
 - Reserva de actividades exclusivas para empadronados.
 - Además, todas las funcionalidades del usuario normal.
- **FUNCIONALIDADES ADMINISTRADOR**
 - Añadir, editar y eliminar noticias.
 - Añadir, editar y eliminar eventos.
 - Visualizar y eliminar los usuarios inscritos a eventos.
 - Editar horarios y precios, habilitar o deshabilitar y especificar si son exclusivas para empadronados las actividades.
 - Visualizar y eliminar las reservas de las actividades.
 - Añadir, editar, eliminar y modificar la reserva de libros.
 - Editar y eliminar cuentas de roles no administrador.



SCRUM

Sprint 1

Descripción

- Fecha comienzo: 25/04/2023
- Fecha final: 17/05/2023
- Objetivo del Sprint:
 - Crear la base de datos MongoDB.
 - Creación del repositorio y de sus consecuentes ramas.
 - Hacer toda la funcionalidad del administrador.
 - Crear el inicio de sesión y registro de los usuarios no administradores.

Contenido de la pila

ID	Nombre	Importancia	Estimación inicial	Cómo probarlo	Notas
#1	Crear repositorio	5	4 horas	Creando repositorio y subiendo proyecto	Meter de colaboradores a todos los miembros del equipo
#2	Crear DDBB	5	1 día	Insertando y eliminando elementos	Tener precaución con las claves foráneas



ID	Nombre	Importancia	Estimación inicial	Cómo probarlo	Notas
#3	Crear .gitignore	2	2 horas	Comprobando que no se suben archivos especificados en el .gitignore	
#4	Crear ramas del repositorio	3	1 hora	Hacer commits y mergear con master	No modificar los mismos archivos en distintas ramas
#5	Administrador noticias	1	2 días	Creando, modificando y eliminando noticias	No preocuparse por parte no administrador
#6	Administrador eventos	1	2 días	Creando, modificando y eliminando eventos	No preocuparse por parte no administrador
#7	Administrador actividades	1	2 días	Modificar datos de las actividades	Se podrá también eliminar sus reservas
#8	Registro no administrador	3	2 días 4 horas	Creando usuarios y comprobando en BBDD	Se deben aplicar tipo empadronados a DNI's en el censo



ID	Nombre	Importancia	Estimación inicial	Cómo probarlo	Notas
#9	Inicio sesión no administrador	3	2 días 4 horas	Accediendo a la parte de usuarios	Ignorar parte administrador
#10	Administrador usuarios	2	2 días	Modificando y eliminando usuarios	Ignorando parte no administrador
#11	Administrador libros	2	1 días	Añadir libros y comprobar en la base de datos. Además probar a eliminarlos.	El administrador podrá reservar libros y dejarlos inhabilitados.
#12	Corregir errores y depurar	5	1 día 4 horas	Probando las funcionalidades creadas hasta el momento.	No subir al repositorio hasta haber comprobado que no dejó de funcionar algo que antes lo hacía.

Acta de reunión de planificación

- Información general:
 - Fecha de realización: 25 abril 2023
 - Número de Sprint: 1
 - Asistentes a la reunión: Alberto Gutiérrez Morán, Miguel González García y Miguel Ángel Pérez López.



- Objetivos de la reunión:
 - Definir tipo de BBDD.
 - Empleo de base de datos MongoDB que es una base de datos NoSQL orientada a documentos.
 - Adaptación a repositorio.
 - Añadir como participantes a Miguel González y Miguel Ángel.
 - Definir tareas para el sprint.
 - Definidas en la pila.
 - Definir la importancia de las tareas y estimación de tiempo.
 - Definidas en la pila.
 - División de las tareas.
 - Alberto Gutiérrez comenzó con la creación de la BBDD.
 - Miguel González comenzó con la adaptación del repositorio.
 - Posteriormente cada miembro que finalice una tarea coge una tarea sin empezar para seguir trabajando siempre teniendo en cuenta la prioridad e importancia de la tarea.

Acta de reunión de revisión

- Información general:
 - Fecha de realización: 17 mayo 2023
 - Número de Sprint 1



- Asistentes a la reunión: Alberto Gutiérrez Morán, Miguel González García.
- Objetivos de la reunión:
 - Qué ha funcionado en el Sprint.
 - Aplicación funcional hasta donde se propuso.
 - Qué mejorar en el siguiente Sprint.
 - Interfaz gráfica de la aplicación.
 - Problemas para progresar en el Sprint.
 - Aprendizaje del nuevo lenguaje de programación.
 - Mucha carga de trabajo externo.
 - Control de errores.

Acta de reunión de retrospectiva

- Información general:
 - Fecha de realización: 17 mayo 2023
 - Número de Sprint 1
 - Asistentes a la reunión: Alberto Gutiérrez Morán, Miguel González García.
- Objetivos de la reunión:
 - Qué salió bien en el último Sprint
 - La estimación de tiempos



- Qué salió mal
 - Estimación de la importancia de las tareas.
- Qué aprendimos
 - Buena estimación de los tiempos de las tareas.
 - Debilidades de cada compañero.
- Qué deberíamos hacer diferente la próxima vez.
 - Dividir mejor las tareas.
 - Mejor estimación de importancia de tareas.

Sprint 2

Descripción

- Fecha comienzo: 18/05/2023
- Fecha final: 02/06/2023
- Objetivo del Sprint:
 - Funcionalidad de los usuarios no administradores
 - Mejorar interfaz gráfica
 - Depurar el código
 - Preparar despliegue

Contenido de la pila

ID	Nombre	Importancia	Estimación inicial	Cómo probarlo	Notas
#11	Modificación de los datos personales	3	2 días	Modificando los datos e iniciar sesión con distinto nombre de usuario o mirar en BBDD si se actualizó	Los campos estén rellenos con los datos actuales del usuario
#12	Reserva de actividades	2	4 días	Reservar y comprobar en BBDD que realmente se hizo	No se podrá reservar una actividad si en el rango de hora elegido ya hay reserva
#13	Inscripción de eventos	2	2 días	Apuntarse y desapuntarse a eventos en los que haya que hacerlo y ver la lista de inscritos	No todos los eventos tienen inscripción
#14	Alquiler de libros	2	3 días	Un libro reservado no se puede reservar por otro usuario	Se debería poder filtrar por bibliotecas y un máximo de un libro por biblioteca. Exclusivo empadronado



ID	Nombre	Importancia	Estimación inicial	Cómo probarlo	Notas
#15	Mejora de interfaz	1	2 días	Navegando por todas las ventanas de la aplicación y viendo que se efectuaron los cambios	Aplicar los mismos estilos en todas las ventanas
#16	Depurar el código	4	4 días	Probando las funcionalidades creadas hasta el momento	No subir al repositorio hasta haber comprobado que no dejó de funcionar algo que antes lo hacía.
#17	Preparar despliegue	5	1 días	Funciona todo correctamente	

Acta de reunión de planificación

- Información general:
 - Fecha de realización: 18 mayo 2023
 - Número de Sprint: 2
 - Asistentes a la reunión: Alberto Gutiérrez Morán, Miguel González García.
- Objetivos de la reunión:
 - Definir tareas para el sprint.



- Definidas en la pila.
- Definir la importancia de las tareas y estimación de tiempo.
- Definidas en la pila.
- División de las tareas.
- Alberto comenzó con la reserva de actividades.
- Miguel González comenzó con la modificación de datos personales.
- Posteriormente cada miembro que finalice una tarea coge una tarea sin empezar para seguir trabajando siempre teniendo en cuenta la prioridad e importancia de la tarea.

Acta de reunión de revisión

- Información general:
 - Fecha de realización: 2 junio 2023
 - Número de Sprint 2
 - Asistentes a la reunión: Alberto Gutiérrez Morán, Miguel González García.
- Objetivos de la reunión:
 - Qué ha funcionado en el Sprint.
 - Aplicación funcional hasta donde se propuso.
 - Qué mejorar en el siguiente Sprint.
 - Añadir nuevas funcionalidades que el cliente desee.



- Depuración del código.
- Problemas para progresar en el Sprint.
- Problemas con la reutilización del código.
- Mucha carga de trabajo externo.

Acta de reunión de retrospectiva

- Información general:
 - Fecha de realización: 2 junio 2023.
 - Número de Sprint 2.
 - Asistentes a la reunión: Alberto Gutiérrez Morán, Miguel González García.
- Objetivos de la reunión:
 - Qué salió bien en el último Sprint:
 - La estimación de tiempos.
 - División de las tareas.
 - Finalización de la aplicación a tiempo.
 - Qué salió mal:
 - Comunicación del equipo de desarrollo.
 - Miguel Ángel Pérez no trabajó.
 - Qué aprendimos:

- Mejor división de tareas que el sprint anterior.
- Qué deberíamos hacer diferente la próxima vez:
 - La comunicación del equipo de desarrollo.

Pruebas de unidad

Prueba de caja blanca

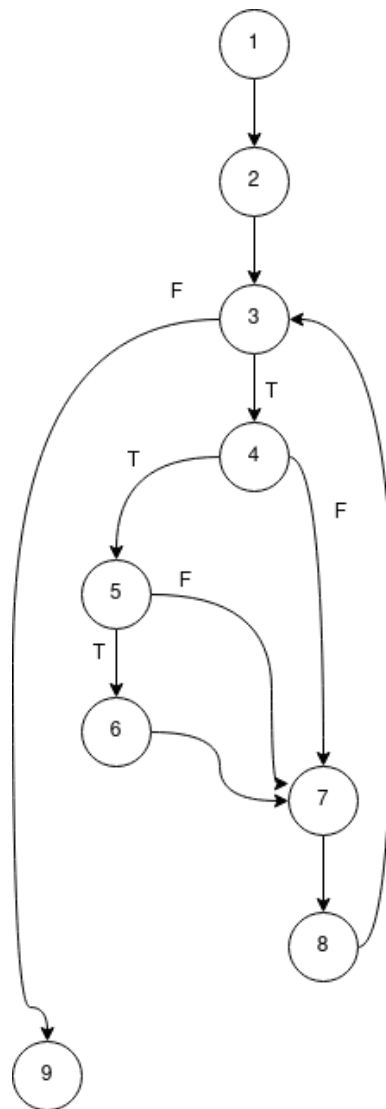
Cálculo de la complejidad ciclomática

- Código extraído de la función `getProximoEvento(data)` situado el directorio `client/src/App.js`.

```
function getProximoEvento(data){  
  const CurrentDate = new Date();  
  let evento = data[0];  
  for(let i=1; i<data.length; i++){  
    if(CurrentDate <= new Date(data[i].fecha) && new Date(data[i].fecha) <  
      new Date(evento.fecha)){  
      evento = data[i];  
    }  
  }  
  
  setEventos(data);  
  setProximoEvento(evento);  
  setLoadingEventos(false);  
}
```



- Grafo de flujo:





- Cálculo de la complejidad ciclomática:
 - $V(G) = R = 4$
 - $V(G) = E - N + 2 = 11 - 9 + 2 = 4$
 - $V(G) = P + 1 = 3 + 1 = 4$
- Conjunto básico de caminos independientes:
 - Camino 1: 1-2-3-9 Aristas nuevas reconocidas: 1-2, 2-3, 3-9
 - Camino 2: 1-2-3-4-7-8-3-9 Aristas nuevas reconocidas: 3-4, 4-7, 7-8, 8-3
 - Camino 3: 1-2-3-4-5-7-8-3-9 Aristas nuevas reconocidas: 4-5, 5-7
 - Camino 4: 1-2-3-4-5-6-7-8-3-9 Aristas nuevas reconocidas: 5-6, 6-7
- Casos de prueba:
 - Caso de prueba camino 1: 1-2-3-9: `data.length = 1;`

Resultados: Retorna el evento más próximo
 - Caso de prueba camino 2: 1-2-3-4-7-8-3-9: `data.length = 2; CurrentDate > new Date(data[i].fecha);`

Resultados: Retorna el evento más próximo
 - Caso de prueba camino 3: 1-2-3-4-5-7-8-3-9: `data.length = 2; CurrentDate <= new Date(data[i].fecha); new Date(data[i].fecha) >= new Date(evento.fecha);`

Resultados: Retorna el evento más próximo
 - Caso de prueba camino 4: 1-2-3-4-5-6-7-8-3-9: `data.length = 2; CurrentDate <= new Date(data[i].fecha); new Date(data[i].fecha) < new Date(evento.fecha);`

Resultados: Retorna el evento más próximo

Prueba de caja negra

Partición de equivalencia

Identificación de las clases de equivalencia

Sección	Condición de entrada	Tipo	Clases válidas		Clases no válidas	
			Entrada	Código	Entrada	Código
1	CurrentDate	Valor específico	Fecha actual	CEV<01>	Año anterior	CENV<01>
					Año posterior	CENV<02>
2	evento	Miembro de un conjunto	Primer miembro del conjunto data	CEV<02>	Otro miembro	CENV<03>
					No hay miembros	CENV<04>
3	data[i].fecha	Miembro de un conjunto	Miembro n.º i del conjunto data	CEV<03>	Otro miembro	CENV<05>
					Fuera de rango	CENV<06>
4	evento.fecha	Valor específico	Fecha del evento	CEV<04>	Mes anterior	CENV<07>
					Mes posterior	CENV<08>

Creación de los casos de prueba

ID CP	Clases de equivalencia	CONDICIONES DE ENTRADA				Resultado esperado
		currentDate	evento	data[i].fecha	evento.fecha	
CP1	CEV<01>, CEV<02>, CEV<03>, CEV<04>	Fecha actual	Primer miembro del conjunto data	Miembro n.º i del conjunto data	Fecha del evento	Ordenación por fecha de los eventos
CP2	CENV<02>, CEV<02>, CEV<03>, CEV<04>	Dos años después	Primer miembro del conjunto data	Miembro n.º i del conjunto data	Fecha del evento	Ordenación incorrecta
CP3	CEV<01>, CENV<03>, CEV<03>, CEV<04>	Fecha actual	Otro miembro que no sea el primero	Miembro n.º i del conjunto data	Fecha del evento	Ordenación incorrecta
CP4	CEV<01>, CENV<04>, CEV<03>, CEV<04>	Fecha actual	No hay miembros en el conjunto data	Miembro n.º i del conjunto data	Fecha del evento	NullPointerException
CP5	CEV<01>, CEV<02>, CENV<05>, CEV<04>	Fecha actual	Primer miembro del conjunto data	Otro miembro que no sea el n.º i	Fecha del evento	Ordenación incorrecta
CP6	CEV<01>, CEV<02>, CENV<06>, CEV<04>	Fecha actual	Primer miembro del conjunto data	Intento de acceso a un miembro inexistente	Fecha del evento	NullPointerException

CP7	CEV<01>, CEV<02>, CEV<03>, CENNV<07>	Fecha actual	Primer miembro del conjunto data	Miembro n.º i del conjunto data	Un mes antes	Ordenación incorrecta
CP8	CEV<01>, CEV<02>, CEV<03>, CENNV<08>	Fecha actual	Primer miembro del conjunto data	Miembro n.º i del conjunto data	Un mes después	Ordenación incorrecta

Controlador de versiones

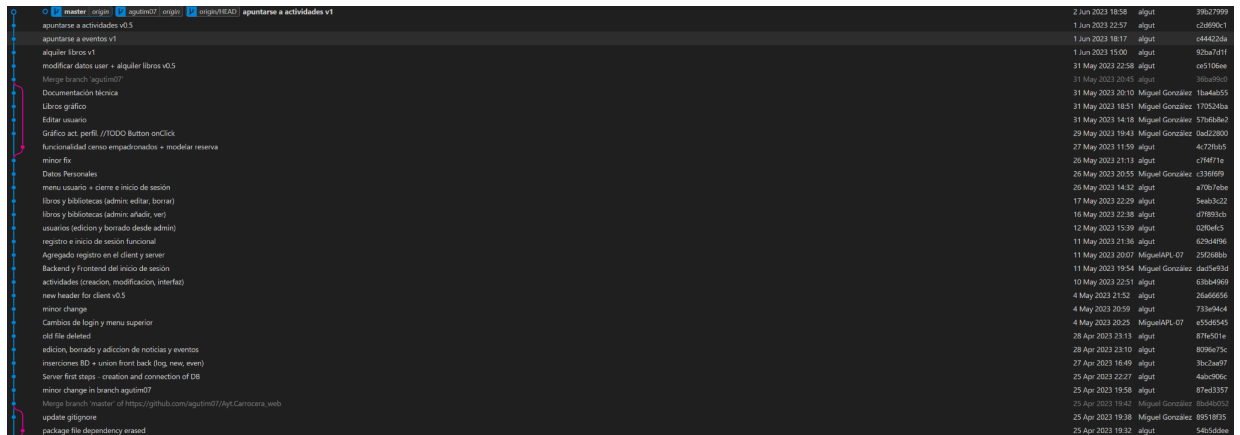
Para controlar las versiones del proyecto software utilizamos el software Git. El repositorio usado es el siguiente: https://github.com/agutim07/Ayt.Carrocera_web

La aplicación la reutilizamos de una existente del Ayuntamiento de Carrocera, desarrollada en prácticas externas por un integrante del equipo, que solo disponía de información visual de los pueblos pero sin ninguna funcionalidad de las requeridas (el backend por ejemplo, se ha hecho de cero). Nosotros como desarrolladores de software le añadimos las funcionalidades descritas en puntos anteriores. En los gráficos de git, se muestra cómo fue iniciada en julio del año 2022 pero la funcionalidad correspondiente a esta práctica se empezó entre los miembros del equipo de desarrollo a finales de abril del 2023 hasta inicios de junio de este año.

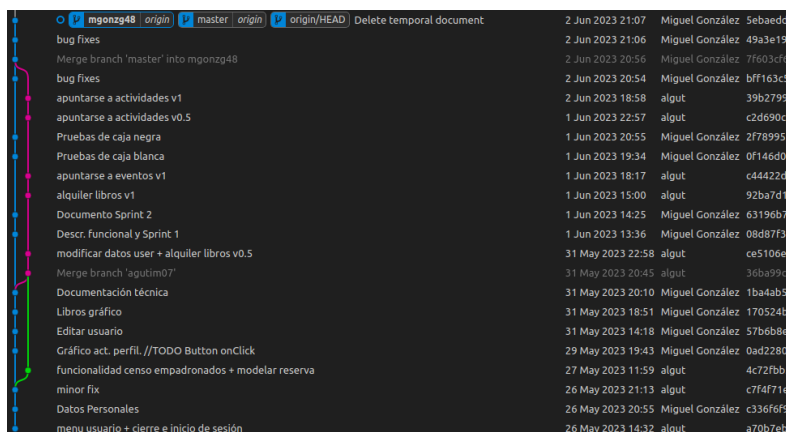
Cada miembro del equipo trabajamos en una rama independiente que, a medida que le añadíamos funcionalidad a la aplicación íbamos uniéndola con la rama master, esto nos permitió tener siempre en la rama principal una versión completamente funcional y sin errores de la aplicación que pudiera originar la unión de ramas.



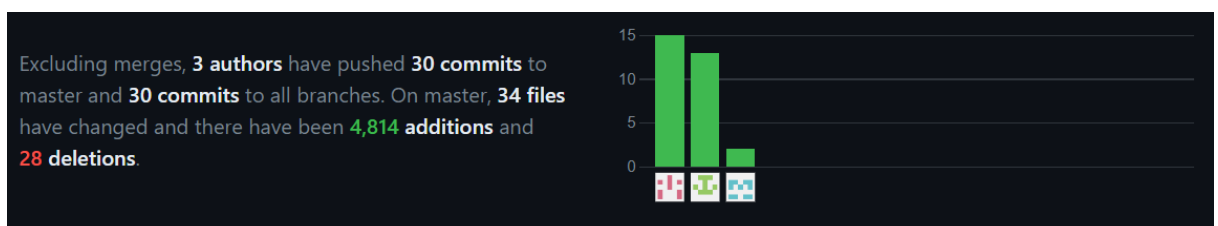
La rama de Alberto Gutiérrez Morán, 'agutim07', generó el siguiente gráfico de commits (un estilo de árbol) mediante la extensión 'Git Graph' del Visual Studio Code:



La rama de Miguel González García, 'MiguelFn99', generó el siguiente gráfico de commits (un estilo de árbol) mediante la extensión 'Git Graph' del Visual Studio Code:



Finalmente añadimos un gráfico, como información visual adicional, de los commits realizados este últimos mes (desde el 2 de Mayo hasta el 2 de Junio) en cada rama. En el gráfico de barras el orden correspondiente es: agutim07, MiguelFn99 y MiguelAPL-07.





Ejecución

El código fuente se encuentra en: https://github.com/agutim07/Ayt.Carrocera_web

Se divide en dos carpetas, "client" equivalente al frontend y "server" equivalente al backend. Para ejecutar correctamente la aplicación ambas se deben de ejecutar por separado paralelamente.

Antes de su ejecución es necesaria la instalación de ciertos paquetes para el correcto funcionamiento de ambas dos entidades:

- Frontend (carpeta 'client'):
 - npm i react
 - npm i axios
 - npm install @emailjs/browser --save
 - npm i dayjs
 - npm install @mui/material @emotion/react @emotion/styled
 - npm install @mui/icons-material

- Backend (carpeta 'server'):
 - npm i express
 - npm i mongoose
 - npm i cors
 - npm i xlsx

Una vez instalados los paquetes, se deben abrir dos terminales paralelamente, posicionar cada una en cada carpeta entidad, y ejecutar tanto el servidor como el cliente con **npm start**. Si todo se ha instalado correctamente el servidor se ejecutará en el puerto 3000, el cliente en el 5000 y la aplicación web funcionará correctamente.

Algunas credenciales para probar la aplicación:

- (Administrador) usuario: adminLuisa | contraseña: carrocera
- (Usuario empadronado) usuario: agm | contraseña: ayt