

Practica is



Componentes del grupo:

- Laura López Burgos (github: i82lobul)
- Cristobal Jesús Moreno Zamora (github: i02mozac)
- Francisco Manuel Velasco Palomero (github: i12vepaf)
- Antonio Gallardo Reina

Dir del repositorio:

<https://github.com/i82lobul/repositorio-compartido.git>

REQUISITOS:

REQUISITOS FUNCIONALES

- RF_01 - Buscar por apellidos: Se realizará una búsqueda por apellidos y se mostrará por pantalla los datos de los usuarios. En caso de no encontrar ningún usuario que corresponda con los apellidos establecidos, se procederá a la activación del RF_02.1.
- RF_02 - Gestión del usuario: Se dará de alta a un nuevo usuario (RF_02.1), modificar los datos de un usuario ya existente previa búsqueda (RF_02.2) o borrar un usuario existente (RF_02.3).
- RF_02.1 - Insertar usuario
- RF_02.2 - Modificar usuario
- RF_02.3 - Borrar usuario
- RF_03 - Ordenar por apellidos: Se ordenará la agenda y se activará RF_04.
- RF_04 - Mostrar por pantalla: Se imprimirá por pantalla el usuario o usuarios en texto plano o en HTML.
- RF_05 - Copia de seguridad: Se copiará a un fichero los datos para que no se pierdan.

REQUISITOS FUNCIONALES

- RNF_01 - Escrito en C++: Esta agenda será desarrollada en lenguaje de alto nivel C++.
- RNF_02 - Se ejecuta en un ordenador con Linux: Utilizamos Linux porque es un SO que nos aporta herramientas para el desarrollo y compilación del programa y es de software libre.
- RNF_03 - Nivel de seguridad cero: No necesitamos ningún sistema de seguridad para nuestro programa
- RNF_04 - Integridad de los datos: Se debe mantener una copia de los datos en un fichero.
- RNF_05 - Volumen máximo de datos de 200 entradas: Para evitar la saturación de datos, el programa mantendrá un límite de 200 entradas.
- RNF_06 - Almacenamiento en texto plano/binario: Se almacenará en un texto plano/binario.

Casos de uso

Caso de uso: BuscarApellido
ID: 1
Breve descripción: Se realizará una búsqueda por apellidos y se mostrará por pantalla los datos de los usuarios. En caso de no encontrar ningún usuario que corresponda con los apellidos establecidos, se procederá a insertar el usuario.
Actores principales: Administrativo
Actores secundarios: Ninguno
Precondiciones: Que haya al menos un registro en la base de datos..
Flujo principal: <ol style="list-style-type: none">1. El caso de uso empieza cuando el Administrativo selecciona “Buscar apellido”2. El sistema pregunta al administrativo el apellido3. El administrativo escribe el apellido correspondiente.4. El sistema busca en la base de datos el apellido que le ha facilitado el administrativo5. Si el sistema encuentra el apellido:<ol style="list-style-type: none">5.1. El sistema muestra todos los datos que se correspondan con el apellido facilitado por el administrativo.
Postcondiciones: Ninguna
Flujo alternativo: <ol style="list-style-type: none">5.a - Si no encuentra ningún paciente informa5.b - Si encuentra más de un paciente los muestra todos.

Caso de uso: InsertarUsuario
ID: 2
Breve descripción: Se dará de alta a un nuevo usuario
Actores principales: Administrativo
Actores secundarios: Ninguno
Precondiciones: Ninguno
Flujo principal: <ol style="list-style-type: none">1. El caso de uso empieza cuando el Administrativo selecciona “Insertar usuario”2. El sistema pregunta al administrativo los datos y si quiere hacerlo favorito3. El administrativo escribe los datos correspondiente y responde4. El sistema guarda en la base de datos al nuevo usuario
Postcondiciones: Los datos del nuevo paciente aparecen en la base de datos y el fichero sigue ordenado.
Flujo alternativo: <ol style="list-style-type: none">2.a - Si ya existe un paciente con ese nombre se sobrescribe.

2.b - Si no está todos los campos obligatorios o hay valores erróneos volvemos a empezar 2.

Caso de uso: ModificarUsuario
ID: 3
Breve descripción: Modificar los datos de un usuario ya existente previa búsqueda
Actores principales: Administrativo
Actores secundarios: Ninguno
Precondiciones: Que el usuario este en la base de datos
Flujo principal: <ol style="list-style-type: none">1. El caso de uso empieza cuando el Administrativo selecciona "Modificar usuario.2. El sistema pregunta al administrativo el apellido del usuario a modificar3. El administrativo le aporta el apellido cuyos datos se modificaran.4. El sistema modifica los datos que se deseen modificar de la base de datos
Postcondiciones: Los datos quedan modificados en la base de datos.
Flujo alternativo: <ol style="list-style-type: none">3. a - Si existen varios pacientes con el mismo apellido se muestra una lista para elegir.4. a - Si no están todos los campos obligatorios o hay errores en los campos se vuelve a empezar 4.

Caso de uso: BorrarUsuario
ID: 4
Breve descripción: Borrar un usuario existente
Actores principales: Administrativo
Actores secundarios: Ninguno
Precondiciones: Que el usuario este en la base de datos
Flujo principal: <ol style="list-style-type: none">1. El caso de uso empieza cuando el Administrativo selecciona "Borrar usuario"2. El sistema pregunta al administrativo el apellido del usuario a borrar3. El administrativo le aporta el apellido del usuario que se borrara.4. El sistema borra el usuario con el apellido aportado.
Postcondiciones: La base de datos no debe mostrar el usuario que se acaba de borrar.
Flujo alternativo: <ol style="list-style-type: none">3. a - Si existen varios pacientes con el mismo apellido se muestra una lista para elegir.4. a - Si no están todos los campos obligatorios o hay errores en los campos se vuelve a empezar 4.

Historias de usuario

La prioridad va de 100 a 400 siendo 400 máxima prioridad

ID: 1	BuscarApellido
Descripción: Como administrativo quiero poder buscar por apellido para ver todos los datos del usuario. Estimación: - Prioridad: 400 Dependencias: 6	
Pruebas de aceptación: <ul style="list-style-type: none">- Insertar un apellido erróneo y comprobar que no aparece ninguno de nuestra base de datos.- Si aparecen más de un cliente con el mismo apellido se mostrarán todos	

ID: 2	InsertarUsuario
Descripción: Como administrativo quiero insertar un usuario para introducir pacientes en la agenda. Estimación: - Prioridad: 300 Dependencias: 5	
Pruebas de aceptación: <ul style="list-style-type: none">- Insertar un usuario que ya se encuentren en nuestra base de datos y que el sistema nos muestre un mensaje de error "usuario ya insertado".- Insertar caracteres no apropiados para un apellido y que aparezca un mensaje de error " no existen apellidos)%\$123..."- Que aparezca correctamente ordenado- Que aparezca insertado en la base de datos	

ID: 3	ModificarUsuario
Descripción: Como administrativo quiero modificar un usuario para hacer cambios en los datos de los pacientes. Estimación: - Prioridad: 200 Dependencias: 2, 5, 1	
Pruebas de aceptación: <ul style="list-style-type: none">- Insertar un usuario que no se encuentre en la base de datos y que envíe un mensaje de error de "usuario no encontrado".- Insertar un dato que no se encuentre en los específicos del cliente y que envíe un mensaje de error "dato a modificar no encontrado".- Que aparezca el usuario modificado en la base de datos	

ID: 4	BorrarUsuario
Descripción: Como administrativo quiero borrar un usuario para que desaparezca de la base	

de datos.

Estimación: -

Prioridad: 200

Dependencias: 5, 1

Pruebas de aceptación:

- Buscar un usuario para borrar que no se encuentre en la base de datos y que envíe un mensaje de error de "usuario no encontrado"