

**DEPARTAMENTO DE**

**CIENCIAS DE LA COMPUTACIÓN**

**INGENIERÍA EN**

**TECNOLOGÍAS DE LA INFORMACIÓN**

*“HISTORIAS DE USUARIO”*

**Integrantes:**

* Panchez Darwin
* Proaño José
* Robalino Cristian

**Tutor:**

Ing. Jenny Ruiz

Historias de usuario para el cargo de utilización del sistema.

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 1 | **Usuario**: Criador | |
| Nombre Historia: Registrar datos de jaula | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Alto |
| Iteración Asignada: 1 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Ingresar el número de jaula, tipo de jaula (engorde, reproducción o crianza) y la capacidad | | |
| Validación:   * Si algún campo está vacío o contiene datos inválidos, se marcan en rojo y se muestra un mensaje indicando el error. * Si los datos son correctos, la jaula se registra y se muestra el mensaje “Registro exitoso”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 2 | **Usuario**: Criador | |
| Nombre Historia: Consultar datos de jaula | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Bajo |
| Iteración Asignada: 1 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Ingresar el número de jaula en un formulario de búsqueda | | |
| Validación:   * Si el número ingresado es inválido o está vacío, se mostrará un mensaje de advertencia. * Si la jaula no existe, se mostrará el mensaje “Jaula no encontrada”. * Si todo es correcto, se muestran los datos de la jaula (número de jaula, tipo de jaula, capacidad). | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 3 | **Usuario**: Criador | |
| Nombre Historia: Editar datos de jaula | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 1 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Acceder a un listado de jaulas, selecciona una y modifica su tipo y capacidad | | |
| Validación:   * Si los datos modificados no cumplen con las restricciones, se marcan en rojo. * Si la validación es exitosa, se guarda la información y se muestra “Edición realizada correctamente”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 4 | **Usuario**: Criador | |
| Nombre Historia: Eliminar jaula | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Bajo |
| Iteración Asignada: 1 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Acceder al listado de jaulas, selecciona una jaula para eliminarla y confirma la acción. | | |
| Validación:   * Si el criador no confirma, no se ejecuta la acción. * Si confirma, se elimina la jaula y se muestra el mensaje “Jaula eliminada con éxito”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 5 | **Usuario**: Criador | |
| Nombre Historia: Registrar raza | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 1 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Ingresar el nombre y la descripción de la raza. | | |
| Validación:   * Si algún campo está vacío o contiene caracteres no válidos, se mostrará un mensaje indicando el error y se marcarán en rojo los campos incorrectos. * Si los datos cumplen con las condiciones y no hay problemas de conexión, se muestra el mensaje “Registro exitoso” y se guarda la raza. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 6 | **Usuario**: Criador | |
| Nombre Historia: Editar raza | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 2 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Acceder a un listado de razas registradas, seleccionar una raza, el sistema muestra sus datos en una forma, permitiendo editar únicamente la descripción. | | |
| Validación:   * Si el campo descripción está vacío, el sistema marcará en rojo el error y no permitirá guardar. * Si los datos son correctos el sistema muestra el mensaje “Raza editada con éxito”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 7 | **Usuario**: Criador | |
| Nombre Historia: Eliminar raza | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Bajo |
| Iteración Asignada: 2 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Acceder al listado de razas, seleccionar una raza específica y aceptar la confirmación de la eliminación. | | |
| Validación:   * Si el criador no confirma la eliminación, el sistema cancela la operación y vuelve al listado. * Si confirma, el sistema elimina la raza y se muestra el mensaje “Raza eliminada con éxito”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 8 | **Usuario**: Criador | |
| Nombre Historia: Registrar datos de conejos | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 2 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Ingresar los siguientes datos: raza, sexo, edad (en meses), peso (en kilogramos) y propósito (Reproducción o Engorde). | | |
| Validación:   * Todos los campos deben estar llenos. * El código debe seguir el formato correcto (único, sin repeticiones, y generado automáticamente). * El sexo debe ser "macho" o "hembra". * La edad debe ser un entero positivo no mayor a 12. * El peso debe ser un número real positivo, con un máximo de 2.5 kg y dos decimales. * El propósito debe ser "Reproducción" o "Engorde". * Si los datos no cumplen con estos criterios, el sistema mostrará los errores en rojo.  Si hay un problema con la base de datos, el sistema informará al usuario que intente más tarde. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 9 | **Usuario**: Criador | |
| Nombre Historia: Consultar datos de conejo | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Bajo |
| Iteración Asignada: 2 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Consultar la información de un conejo específico mediante el código del conejo, mostrando el código, sexo, edad, peso y propósito. | | |
| Validación:   * El campo de búsqueda no debe estar vacío. * El código debe cumplir con el formato: primera letra de la raza + número secuencial de 3 dígitos * Si el formato es incorrecto, el sistema borra el dato ingresado y muestra un mensaje indicando el formato válido. * Si el código es correcto pero no se encuentra ningún conejo, el sistema muestra un mensaje informando que no se encontraron registros. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 10 | **Usuario**: Criador | |
| Nombre Historia: Filtrar por razas de conejo | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Bajo |
| Iteración Asignada: 2 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Filtrar conejos según su raza, mostrando un listado de todos los conejos de esa raza, junto al código, sexo, edad, peso y propósito. | | |
| Validación:   * El campo de búsqueda no debe estar vacío. * El nombre de raza debe contener solo letras. * Si el formato no es válido, el sistema borra el valor ingresado e informa que solo se permiten letras. * Si no existen conejos asociados a esa raza, el sistema muestra un mensaje indicando que no se encontraron registros. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 11 | **Usuario**: Criador | |
| Nombre Historia: Editar datos de conejos | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 3 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Seleccionar un conejo y modifica los datos necesarios: raza, código, sexo, edad, peso y propósito. | | |
| Validación:   * Raza: debe seleccionarse de una lista predefinida. * Código: debe seguir el formato de primera letra de la raza y un número secuencial de 3 dígitos. * Sexo: debe ser "macho" o "hembra". * Edad: debe ser un número entero positivo, no mayor a 12 meses. * Peso: debe ser un número real positivo, con hasta dos decimales y no mayor a 2.5 kg. * Propósito: debe ser "Reproducción" o "Engorde". * Ningún campo debe estar vacío. * Si un dato no es válido, el sistema resalta el campo en rojo. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 12 | **Usuario**: Criador | |
| Nombre Historia: Eliminar conejo | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Bajo |
| Iteración Asignada: 3 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Acceder al listado de conejos, seleccionar uno y aceptar la confirmación de la eliminación. | | |
| Validación:   * Si no confirma, el sistema cancela la eliminación y mantiene el conejo en el listado. * Si confirma, el conejo se elimina permanentemente. * Se muestra el mensaje “Conejo eliminado con éxito” tras cada eliminación exitosa. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 13 | **Usuario**: Criador | |
| Nombre Historia: Registrar datos de monta | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 3 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Seleccionar una coneja apta (sexo hembra, más de 4 meses) para ingresar la fecha de monta y el sistema calcula automáticamente la fecha de parto (30 días después). | | |
| Validación:   * Solo conejas con más de 4 meses y sexo hembra. * No debe existir una monta activa para la coneja en el mismo rango. * Si el registro es exitoso se debe mostrar el mensaje “ Datos registrados con éxito”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 14 | **Usuario**: Criador | |
| Nombre Historia: Eliminar parto | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 3 | | |
| Programador Responsable: Cristian Robalino | | |
| Descripción:   * Seleccionar una coneja del listado que tenga fecha de monta y fecha de parto registradas, y confirmar la eliminación del registro de parto si es necesario. | | |
| Validación:   * Si el criador confirma la eliminación, el sistema borra los datos de monta y parto, mostrando un mensaje de éxito “Fecha eliminada con éxito”. * Si no confirma, se regresa al listado sin realizar cambios. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 15 | **Usuario**: Criador | |
| Nombre Historia: Asignar Jaula a conejo | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 3 | | |
| Programador Responsable: Darwin Panchez | | |
| Descripción:   * Acceder al listado interactivo de jaulas sin asignación y al listado de conejos sin asignar, seleccionar una jaula y uno o varios conejos (hasta completar la capacidad máxima), confirmar la asignación y registrar la relación en el sistema. | | |
| Validación:   * Si no confirma la asignación, el sistema cancela el proceso y retorna a los listados. * Si confirma, se guarda la asignación y se muestra “Asignación realizada con éxito”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 16 | **Usuario**: Criador | |
| Nombre Historia: Controlar alimentación | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 2 | | |
| Programador Responsable: Darwin Panchez | | |
| Descripción:   * Acceder al listado de conejos con jaula asignada y dieta calculada (2 % del peso corporal, desglosado en heno, hierba y balanceado), seleccionar los conejos a los que se les suministró el alimento y registrar la alimentación con fecha y hora. | | |
| Validación:   * Al guardar, validar que al menos el 90 % de los conejos hayan sido seleccionados; si no, mostrar mensaje de advertencia para completar mínimo el 90 %. * Tras un registro exitoso, mostrar “Alimentación registrada con éxito”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 17 | **Usuario**: Criador | |
| Nombre Historia: Controlar vacunación y desparasitación | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 2 | | |
| Programador Responsable: Darwin Panchez | | |
| Descripción:   * Acceder al listado de conejos con jaula asignada, seleccionar uno o varios, indicar qué vacunas (mixomatosis y/o VHD) se aplicaron y si se realizó la desparasitación, validar intervalos mínimos (1 año entre vacunas, 1 mes entre desparasitaciones) y guardar el registro con fecha. | | |
| Validación:   * Validar que se haya seleccionado al menos un conejo y al menos una actividad (vacuna o desparasitación); en caso contrario, mostrar mensaje de error. * Tras registro exitoso, mostrar “Control de vacunación y/o desparasitación registrado con éxito. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 18 | **Usuario**: Criador | |
| Nombre Historia: Controlar crecimiento | | |
| Prioridad en negocio: Alta | | **Riesgo en desarrollo**: Media |
| Iteración Asignada: 2 | | |
| Programador Responsable: Darwin Panchez | | |
| Descripción:   * Acceder al listado de conejos (con datos de jaula, edad, peso y últimos controles), actualizar la edad de todos automáticamente (calculada desde la fecha de registro) y luego, para los conejos seleccionados, ingresar un valor de incremento de peso y registrar la nueva masa, respetando el máximo de 2.5 kg. | | |
| Validación:   * Validar que al menos un conejo esté seleccionado para el ajuste de peso; si no, mostrar mensaje de error. * Validar que el incremento no supere el límite de peso; tras éxito, mostrar “Crecimiento registrado con éxito”. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 19 | **Usuario**: Criador | |
| Nombre Historia: Iniciar Sesión | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Bajo |
| Iteración Asignada: 3 | | |
| Programador Responsable: José Proaño | | |
| Descripción:   * Desplegar un formulario con dos campos: usuario y contraseña. El actor introduce sus credenciales en los campos correspondientes y presiona el botón “Ingresar”. El sistema recibe los datos y procede a validarlos contra la información almacenada en la base de datos. | | |
| Validación:   * Si el campo de usuario o contraseña está vacío, se marca el campo vacío en rojo y se muestra el mensaje:  “Debe ingresar usuario y contraseña”. * Si el actor ingresa credenciales que no coinciden con las registradas, se muestra el mensaje:  “Usuario o contraseña no registrados”. * Si el actor falla más de 4 veces ingresando credenciales inválidas, al quinto intento se muestra el mensaje:  “Has excedido el número máximo de intentos. Vuelve a intentarlo en 24 horas.” * Si la base de datos no está disponible, se muestra el mensaje:  “Error de conexión. Intenta nuevamente más tarde.” * Si las credenciales son correctas, el sistema muestra “Bienvenido, <NombreUsuario>” y permite el acceso al aplicativo. * El tiempo de validación no debe exceder los 3 segundos. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 20 | **Usuario**: Criador | |
| Nombre Historia: Generar reporte de alimentación | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 3 | | |
| Programador Responsable: José Proaño | | |
| Descripción:   * Generar reportes de alimentación de los conejos aplicando filtros por rango de fechas y raza, para obtener información detallada sobre la dieta suministrada y facilitar el seguimiento nutricional y la toma de decisiones en la gestión de la crianza. | | |
| Validación:   * Si la fecha de fin es anterior a la de inicio, el sistema muestra un mensaje de error. * Si no se selecciona al menos una raza, el sistema solicita seleccionar una. * Si no hay datos que coincidan con los filtros, se informa que no se encontraron resultados. * Si los filtros son válidos y hay datos disponibles, se genera una vista previa del reporte y se ofrece la opción de descargarlo en PDF. | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 21 | **Usuario**: Criador | |
| Nombre Historia: Generar reporte de vacunación | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 3 | | |
| Programador Responsable: José Proaño | | |
| Descripción:   * Generar reportes de vacunación de los conejos aplicando filtros por rango de fechas y raza, para obtener un documento detallado que me permita verificar qué vacunas se han aplicado y en qué fechas, facilitando el control sanitario de la crianza. | | |
| Validación:   * Si la fecha de fin es anterior a la de inicio, muestra “La fecha de fin no puede ser anterior a la fecha de inicio. * Si no hay al menos una raza seleccionada, muestra “Debe elegir al menos una raza para generar el reporte.” * Si no hay datos que coincidan con los filtros, muestra “No se encontraron datos para los filtros seleccionados.” * Si los filtros son válidos y hay datos, genera vista previa y ofrece “Exportar a PDF.” | | |

| Historia de Usuario | | |
| --- | --- | --- |
| Número: 22 | **Usuario**: Criador | |
| Nombre Historia: Generar reporte de desparasitación | | |
| Prioridad en negocio: Media | | **Riesgo en desarrollo**: Medio |
| Iteración Asignada: 3 | | |
| Programador Responsable: José Proaño | | |
| Descripción:   * Generar reportes de desparasitación de los conejos aplicando filtros por rango de fechas y raza, para disponer de un documento claro con las fechas en las que cada animal fue desparasitado y facilitar el control sanitario de la crianza. | | |
| Validación:   * Si la fecha de fin es anterior a la de inicio, muestra “La fecha de fin no puede ser anterior a la fecha de inicio.” * Si no hay al menos una raza seleccionada, muestra “Debe elegir al menos una raza para generar el reporte.” * Si no hay datos que coincidan con los filtros, muestra “No se encontraron datos para los filtros seleccionados.” * Si los filtros son válidos y existen datos, genera la vista previa del reporte y ofrece la opción “Exportar a PDF.” | | |