**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**



**TEMA:**

Historias de usuario, historias técnicas y product backlog

**ESTUDIANTE:**

Katty Moyano (7387)

**ASIGNATURA:**

Aplicaciones Informáticas II

**SEMESTRE-PARALELO:**

8° Software “1”

**INGENIERO:**

Ing. Julio Santillán

**FECHA DE ENTREGA:** 14/10/2025

**Introducción**

El presente documento recopila el conjunto de Historias de Usuario e Historias Técnicas (Enablers) correspondientes al desarrollo del sistema HerbaProgram, un módulo especializado para la gestión del Herbario ESPOCH, implementado sobre la plataforma Odoo 17. Estas historias constituyen la base del Product Backlog del proyecto y fueron elaboradas bajo la metodología ágil Scrum, priorizando la entrega incremental de valor y la trazabilidad entre los requisitos funcionales, técnicos y de negocio.

Cada historia refleja la interacción de los actores con el sistema, describiendo los objetivos, la motivación y las funcionalidades esperadas, mientras que las historias técnicas detallan los requerimientos de infraestructura, seguridad, rendimiento y mantenimiento que garantizan la calidad del producto. En conjunto, este documento traduce los requisitos del software a un formato operativo y verificable, asegurando que el desarrollo del sistema cumpla con estándares de eficiencia, accesibilidad, interoperabilidad y seguridad propios de los sistemas informáticos institucionales modernos.

**Desarrollo**

**Historias de Usuario**

* **Historia Usuario 1**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-01 | **Título:** Gestión de especímenes |
| **Descripción:** | |
| Como Encargado del Herbario, quiero crear, leer, actualizar y desactivar (eliminación lógica) registros de especímenes (taxonomía, colector, fecha, coordenadas, notas, imágenes) para mantener la colección digitalizada y auditada. | |
| **Estimación:** 40 | **Dependiente de:** HT-01 (entorno Odoo), HT-02 (BD) |
| **Prioridad:** Alta |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * Al registrar un nuevo espécimen, se genera automáticamente un código único (CHEP-XXXXXXX) no repetido. * Si se intenta guardar un registro con campos obligatorios vacíos, el sistema muestra mensajes claros: “Debe ingresar el nombre científico y la familia antes de guardar.” * El sistema valida que la fecha de recolección no sea futura y que las coordenadas geográficas estén dentro de un rango válido (latitud entre -90 y 90, longitud entre -180 y 180). * Si ocurre un error de conexión o base de datos, el usuario visualiza el mensaje: “No se pudo guardar el registro. Intente nuevamente o contacte al administrador.” * En la eliminación, el registro se marca como inactivo (eliminación lógica) y se conserva en el historial. * Cada modificación registra automáticamente el usuario, fecha y campo alterado, visible en un historial de auditoría. * Guardado exitoso (HTTP 200/201) y registro disponible en listados en < 3seg. | |

* **Historia Usuario 2**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-02 | **Título:** Gestión de imágenes y multimedia |
| **Descripción:** | |
| Como Encargado del Herbario, quiero adjuntar una o varias imágenes a un espécimen y los usuarios poder verlas en galería para identificación y documentación. | |
| **Estimación:** 24 | **Dependiente de:** HU-G1, HT-06 (almacenamiento / backups) |
| **Prioridad:** Media-Alta |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * El sistema permite subir máximo de 1 a 5 imágenes por espécimen, en formatos JPG o PNG. * Cada archivo se valida antes de subirlo: si supera 10 MB, se muestra el error “El archivo excede el tamaño permitido.” * Durante la subida, se muestra una barra de progreso y el usuario puede cancelar la carga. * Si ocurre un fallo durante la carga (por conexión o formato no soportado), se muestra el mensaje “Error al cargar la imagen. Intente nuevamente.” * Las imágenes se almacenan en alta calidad y se generan miniaturas automáticamente (thumbnail ≤200 KB). * Al visualizar, la galería carga todas las imágenes correctamente y las miniaturas se muestran en <5 s. | |

* **Historia Usuario 3**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-03 | **Título:** Búsqueda y filtros avanzados |
| **Descripción:** | |
| Como Usuario/Encargado, quiero buscar especímenes por diversos filtros como: nombre científico, familia, coordenadas, fecha y otros filtros para recuperar registros rápidamente. | |
| **Estimación:** 24 | **Dependiente de:** HT-07 (optimización / índices) y HU-01 (gestión de registros) |
| **Prioridad:** Alta |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * La búsqueda simple devuelve resultados en menos de 3 segundos con hasta 10 000 registros. * Si el usuario no ingresa ningún criterio, el sistema muestra un aviso: “Ingrese al menos un parámetro de búsqueda.” * Filtros múltiples (por familia, colector y año) deben combinarse correctamente (operador lógico “AND”). * Si la búsqueda no devuelve resultados, mostrar “No se encontraron coincidencias.” * Ante error de servidor, mostrar mensaje controlado y registrar evento en logs. * Ordenamiento funcional por nombre, fecha o familia. | |

* **Historia Usuario 4**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID: HU-04** | **Título:** Gestión completa de usuarios y permisos |
| **Descripción:** | |
| Como Administrador TI / Encargado del Herbario, quiero crear, modificar,  eliminar (borrado lógico) y asignar roles diferenciados (Administrador TI,  Encargado Herbario, Usuario-Investigador) para controlar accesos de forma  granular y mantener seguridad del sistema. | |
| **Estimación:** 16 | **Dependiente de:** HT-05 (seguridad / autenticación) |
| **Prioridad:** Alta |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * Solo un Administrador puede crear o eliminar usuarios. * Al crear un nuevo usuario, el sistema valida que el correo electrónico no esté repetido. * Si el correo ya existe, mostrar “Este usuario ya está registrado.” * Las contraseñas deben tener mínimo 8 caracteres, incluir mayúscula, número y símbolo; si no cumple, mostrar mensaje de validación. * Al desactivar un usuario, no puede acceder al sistema (mensaje: “Su cuenta ha sido deshabilitada.”). * Todo intento fallido de inicio de sesión se registra con IP y hora. * Los roles deben restringir funciones (por ejemplo, el usuario “Encargado” no puede crear otros usuarios). * Al crear usuario, validar correo único con dominio @espoch.edu.ec. * Si correo existe, mostrar "Este usuario ya está registrado." | |

* **Historia Usuario 5**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-05 | **Título:** Reportes PDF y exportaciones |
| **Descripción:** | |
| Como Encargado, quiero generar reportes PDF (con filtros y selección de campos) para entregables e investigaciones. | |
| **Estimación:** 12 | **Dependiente de:** HU-G3 (filtros) |
| **Prioridad:** Baja |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * El reporte debe incluir encabezado institucional, fecha de generación y firma digital del sistema. * Si el filtro genera más de 5000 registros, mostrar advertencia “El reporte será dividido en varias páginas o enviado por correo electrónico.” * Si no se selecciona ningún campo o filtro, mostrar “Debe elegir al menos un criterio para generar el reporte.” * El archivo PDF se descarga correctamente y su tamaño no excede 10 MB. * Ante error de conexión, mostrar “No se pudo generar el reporte. Intente nuevamente.” * Reportes generados en <60 segundos. | |

* **Historia Usuario 6**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-06 | **Título:** Estadísticas y visualizaciones |
| **Descripción:** | |
| Como Usuario/Encargado, quiero ver gráficos y tablas (familia, especies, por ubicación) que incluso se puede aplicar filtros para obtener insights. | |
| **Estimación:** 14 | **Dependiente de:** HU-G1, HU-03, HT-07 (consultas optimizadas) |
| **Prioridad:** Media |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * Los gráficos se actualizan automáticamente al cambiar los filtros. * Si no hay datos suficientes, mostrar mensaje “No existen registros para graficar.” * La carga del gráfico no debe exceder 5 seg. * Las barras o sectores reflejan valores exactos respecto a la base de datos. * Si ocurre un error de conexión con el servidor de datos, mostrar mensaje y recargar. | |

* **Historia Usuario 7**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-07 | **Título:** Geolocalización y Mapas |
| **Descripción:** | |
| Como Usuario/Encargado, quiero ver especímenes en mapa interactivo que se puede filtrar por especie (OpenStreetMap / Google Maps) para localizar colectas. | |
| **Estimación:** 18 | **Dependiente de:** HU-G1, HU-03, HT-10 (API mapas) |
| **Prioridad:** Media |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * El mapa se carga correctamente en menos de 5 segundos. * Los marcadores se colocan de acuerdo a las coordenadas registradas. * Si una coordenada está vacía o fuera de rango, no se muestra marcador y aparece “Coordenada inválida.” * El usuario puede hacer zoom, arrastrar el mapa y ver información al hacer clic en un marcador. * Ante fallo de conexión con el servicio de mapas, mostrar “Error al cargar mapa. Verifique su conexión.” | |

* **Historia Usuario 8**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-08 | **Título:** Códigos QR y compartición |
| **Descripción:** | |
| Como Encargado, quiero generar QR por registro para acceso rápido desde móviles y ver la información relevante del espécimen. | |
| **Estimación: 6** | **Dependiente de:** HU-G1, HT-04 (Nginx/URL estable) |
| **Prioridad: Media-Baja** |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * Cada QR redirige a la ficha del espécimen con autenticación o acceso público controlado. * Si el código QR está dañado o no existe, se muestra “El código QR no es válido o el registro fue eliminado.” * Generación de QR tarda menos de 2 segundos. * Se puede descargar o imprimir el código QR en formato PNG o PDF. * Acceso vía QR debe tener trazabilidad: registrar cuántas veces se escanea. | |

* **Historia Usuario 9**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HU-09 | **Título:** Historial de modificaciones |
| **Descripción:** | |
| Como Encargado del Herbario o Administrador TI, quiero consultar el historial  completo de cambios de cualquier registro botánico para tener trazabilidad  científica y auditoría de quién, cuándo y qué modificó en cada espécimen. | |
| **Estimación:** 6 | **Dependiente de:** HU-01 (gestión de registros) |
| **Prioridad:** Media-Alta |
| **Reverso** | |
| **Pruebas de aceptación:** | |
| * El historial se muestra en pestaña dedicada solo para estos. * Entradas ordenadas cronológicamente (más reciente primero). * Cada entrada muestra: fecha/hora, usuario, campo modificado, valor anterior, valor nuevo, especimen, etc. * Si no hay modificaciones: "Este registro no ha sido modificado desde su creación." * Formato de fecha legible: "DD/MM/AAAA HH:MM:SS". * Timeline visual con iconos diferenciados: creación, modificación, eliminación. * Carga de historial < 2 segundos para registros con hasta 50 modificaciones. * Ante error de consulta, mostrar: "Error al cargar historial. Intente nuevamente." | |

**Historias técnicas**

* **Historias técnicas 1**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-01 | **Título:** Preparar entorno base: Odoo 17 para desarrollo |
| **Descripción:** | |
| Instalar, configurar y documentar el entorno de desarrollo en Odoo 17 Community. Esto incluye dependencias de Python, librerías requeridas, creación de entorno virtual, y verificación del correcto despliegue del servidor local.  Es la base para todos los desarrollos posteriores, garantizando uniformidad entre los equipos y evitando errores de compatibilidad. | |
| **Estimación:** 8 | **Dependiente de:** Ninguna |
| **Prioridad:** Alta |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Odoo 17 inicia correctamente en entorno local (puerto 8069) sin errores críticos en el log. * El módulo base “herbaprogram\_base” se instala exitosamente y aparece en la interfaz de Odoo. * En caso de error de dependencia o librería ausente, el sistema debe mostrar mensaje de instalación fallida y registrar el paquete faltante en un archivo de logs (missing\_requirements.log). * Todos los desarrolladores deben poder clonar el entorno y ejecutarlo con un solo comando (make setup o docker-compose up). * Documentación de instalación reproducible con pasos verificados por un segundo miembro del equipo. | |

* **Historias técnicas 2**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-02 | **Título:** Configuración base de PostgreSQL y políticas de respaldo |
| **Descripción:** | |
| Crear y configurar la base de datos PostgreSQL dedicada para el módulo HerbaProgram. Incluir usuario exclusivo con permisos mínimos, estructura inicial (esquemas, índices base) y scripts de backup/restore automáticos.  Garantiza seguridad, integridad y continuidad de los datos. | |
| **Estimación:** 10 | **Dependiente de:** HT-01 |
| **Prioridad:** Alta |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Base de datos creada con nombre herbaprogram\_db y propietario odoo\_user. * Backups automáticos generados de forma semanal o mensual y verificados manualmente una vez por mes. * Cada backup debe completarse sin errores y pesar < 1 GB para 10k registros simulados. * En caso de error en la ejecución del backup, debe enviarse alerta al correo del administrador (asunto: “Fallo en backup semanal”). * Restauración validada en entorno de pruebas en < 30 minutos. * Acceso remoto deshabilitado excepto desde el contenedor Odoo (seguridad de capa de red). | |
| **Tareas:** | |
| * Configurar PostgreSQL. * Crear usuarios, roles y permisos. * Programar backups automáticos y notificaciones. | |

* **Historias técnicas 3**

|  |  |  |
| --- | --- | --- |
| **Anverso** | | |
| **ID:** HT-03 | **Título:** Contenerización completa con Docker | |
| **Descripción:** | | |
| Implementar arquitectura basada en contenedores Docker para Odoo, PostgreSQL y Nginx. La contenerización mejora la portabilidad, reproducibilidad y despliegue en entornos controlados. | | |
| **Estimación:** 12 | **Dependiente de:** HT-01, HT-02 | |
| **Prioridad: Alta** |
| **Reverso** | | |
| **Criterios de aceptación:** | | |
| * docker-compose up crea correctamente los contenedores en menos de 2 minutos. * Los servicios odoo, db, nginx inician sin errores (exit code 0). * Los contenedores deben reiniciarse automáticamente en caso de fallo (restart: always). * Si el puerto 8069 está ocupado, mostrar advertencia: “Puerto en uso, modifique el archivo .env.” * Los volúmenes persisten los datos tras reinicio del contenedor. * Documentación actualizada sobre el flujo de despliegue local y producción. | | |
| **Tareas:** | | **Pruebas:** |
| * Crear Dockerfiles optimizados. * Definir docker-compose.yml. * Probar reconstrucción en entorno limpio. | | * Integración: conexión entre contenedores. * Sistema: persistencia tras reinicio. |

* **Historias técnicas 4**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-04 | **Título:** Configuración de Nginx con HTTPS y Proxy Reverso |
| **Descripción:** | |
| Implementar servidor Nginx para servir Odoo mediante proxy inverso con HTTPS (certificados Let’s Encrypt). Esto protege la comunicación y mejora el rendimiento mediante cacheo y compresión GZIP. | |
| **Estimación:** 6 | **Dependiente de:** HT-03 |
| **Prioridad:** Alta |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Acceso al sistema a través de https://herbaprogram.local sin advertencias de seguridad. * Certificado SSL válido, renovable automáticamente (Certbot). * Redirección obligatoria de HTTP → HTTPS. * En caso de fallo de certificación, el sistema debe seguir operativo en modo HTTP temporal, registrando alerta. * Tiempos de respuesta < 300 ms para peticiones estáticas. | |
| **Tareas:** | |
| * Configurar Nginx como proxy. * Implementar Certbot y auto-renovación. * Ajustar políticas de cabecera y cache. | |

* **Historias técnicas 5**

|  |  |  |
| --- | --- | --- |
| **Anverso** | | |
| **ID:** HT-05 | **Título:** Seguridad y autenticación | |
| **Descripción:** | | |
| Aplicar políticas de autenticación seguras (hash PBKDF2 o bcrypt), gestión de sesiones, protección CSRF, control de permisos y timeout de inactividad. También incluye pruebas OWASP básicas. | | |
| **Estimación:** 16 | **Dependiente de:** HT-01, HU-G4 | |
| **Prioridad:** Alta |
| **Reverso** | | |
| **Criterios de aceptación:** | | |
| * Contraseñas cifradas con algoritmos seguros (bcrypt o SHA-256 con salt). * Sesiones expiran automáticamente tras 30 minutos de inactividad. * Intentos fallidos de login superiores a 5 bloquean temporalmente la cuenta. * Si ocurre intento de acceso no autorizado, mostrar mensaje: “Acceso denegado. No tiene permisos para esta acción.” * Todos los eventos de seguridad se registran en archivo security\_audit.log con IP, fecha y usuario. * Escaneo OWASP ZAP sin vulnerabilidades críticas. | | |
| **Tareas:** | | **Pruebas:** |
| * Configurar políticas de login. * Implementar CSRF tokens. * Desarrollar logs de seguridad. | | * Unitarias: validación de contraseña. * Integración: sesiones y expiración. * Sistema: prueba de intrusión y bloqueo. |

* **Historias técnicas 6**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-06 | **Título:** Estrategia de almacenamiento de imágenes (local vs nube) |
| **Descripción:** | |
| Investigar, comparar e implementar estrategia óptima de almacenamiento: local (servidor institucional) o externo (S3, MinIO). Incluye consultar el rendimiento, costos, seguridad y backup. | |
| **Estimación:** 10 | **Dependiente de:** HT-02, HT-03 |
| **Prioridad:** Alta |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Informe comparativo con métricas: velocidad promedio (ms), disponibilidad (%), costo mensual estimado y complejidad. * Prueba de subida/lectura de 100 imágenes de 2 MB con latencia promedio < 300 ms. * Si el servicio externo no responde, el sistema debe cambiar automáticamente a almacenamiento local temporal. * Backups automáticos de imágenes ejecutados semanalmente. | |

* **Historias técnicas 7**

|  |  |  |
| --- | --- | --- |
| **Anverso** | | |
| **ID:** HT-07 | **Título:** Optimización de rendimiento e índices | |
| **Descripción:** | | |
| Analizar consultas SQL, identificar cuellos de botella y aplicar índices y caching (Redis). Garantiza rendimiento y tiempos de respuesta aceptables. | | |
| **Estimación:** 12 | **Dependiente de:** HT-02, HU-G3 | |
| **Prioridad:** Alta |
| **Reverso** | | |
| **Criterios de aceptación:** | | |
| Búsquedas complejas se ejecutan en < 3 segundos (95° percentil).  Reducción de CPU en servidor DB en 30% comparado con prueba inicial.  Creación de índices en campos “nombre científico”, “familia” y “colector”.  Si un índice genera error o bloqueo, rollback automático y alerta en logs. | | |
| **Tareas:** | | **Pruebas:** |
| * Análisis de consultas lentas. * Creación de índices y optimización de JOINs. * Configurar Redis cache. | | * Unitarias: verificación de consultas. |

* **Historias técnicas 8**

|  |  |  |
| --- | --- | --- |
| **Anverso** | | |
| **ID:** HT-08 | **Título:** Framework de pruebas | |
| **Descripción:** | | |
| Definir y configurar entorno de pruebas para garantizar calidad del código. Se usarán PyTest, Odoo Testing Framework y Playwright para pruebas de interfaz. | | |
| **Estimación:** 12 | **Dependiente de:** Ninguna | |
| **Prioridad:** Alta |
| **Reverso** | | |
| **Criterios de aceptación:** | | |
| * Ejecución completa de suite de pruebas en < 5 minutos. * 100% de las funciones críticas tienen test unitario. * En caso de fallo, logs detallados generados en /reports/tests. * Cobertura mínima del módulo principal: 70%. | | |
| **Tareas:** | | **Pruebas:** |
| * Configurar entorno PyTest. * Crear tests de integración CRUD. * Configurar Playwright para UI. | | * Unitarias: validación de funciones. * Integración: API y DB. * Sistema: flujo completo CRUD. |

* **Historias técnicas 9**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-09 | **Título:** Integración de mapas (OpenStreetMap vs Google Maps) |
| **Descripción:** | |
| Investigar y configurar API de mapas interactivos. Evaluar uso de OpenStreetMap (open source) frente a Google Maps (licenciado). | |
| **Estimación:** 8 | **Dependiente de:** HT-03, HU-G7 |
| **Prioridad:** Media |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Informe comparativo con parámetros: costo, rendimiento, facilidad de integración. * Prototipo funcional con mapa dinámico. * Latencia promedio de carga del mapa < 5 segundos. * En caso de error de conexión a API, mensaje visible y log registrado. | |
| **Tareas:** | |
| * Prototipo Leaflet (OSM). * Evaluar integración Google Maps. * Documentar resultados. | |

* **Historias técnicas 10**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-10 | **Título:** Generación y gestión de códigos QR |
| **Descripción:** | |
| Desarrollar el sistema para generar, almacenar y servir códigos QR únicos asociados a los especímenes registrados.  Los QR deben permitir acceso directo a la ficha de cada espécimen, ya sea pública o autenticada, garantizando seguridad y trazabilidad de los accesos. | |
| **Estimación: 6** | **Dependiente de:** HT-04 |
| **Prioridad: Media** |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Cada registro genera un QR único e irrepetible, vinculado al código del espécimen. * El QR debe crearse en menos de 2 segundos e incluir un hash único en su URL para evitar manipulación. * Al escanear el QR:   + Si el registro existe → muestra ficha correctamente (HTTP 200).   + Si no existe o está inactivo → muestra mensaje “Registro no disponible o desactivado.” * Los QR se almacenan en formato PNG 300 dpi y se regeneran automáticamente si cambia la URL base. * En caso de fallo al generar el QR (error de librería o ruta no accesible), registrar el evento en qr\_error.log. | |
| **Tareas:** | |
| * Implementar librería qrcode en Python. * Crear endpoint seguro /specimen/qr/<id>. * Configurar carpeta de almacenamiento. * Verificar compatibilidad con impresión en PDF. | |

* **Historias técnicas 11**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-11 | **Título:** Políticas de backup y recuperación de información |
| **Descripción:** | |
| Implementar un sistema automatizado de copias de seguridad y restauración para base de datos, archivos e imágenes del sistema.  El objetivo es garantizar continuidad operativa y recuperación ante pérdida de datos o fallos del sistema. | |
| **Estimación:** 8 | **Dependiente de:** HT-02, HT-06 |
| **Prioridad:** Media-Alta |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Backups automáticos generados cada semana sin intervención manual. * En caso de fallo, se envía alerta por correo y notificación en el panel de administrador. * Restauración verificada exitosamente al menos una vez por sprint en entorno de pruebas. * El proceso completo de recuperación no debe exceder 60 minutos. * Los respaldos se almacenan cifrados (AES-256) en ubicación segura y replicada. | |
| **Tareas:** | |
| * Crear scripts bash/Python de backup. * Configurar cron y logs. * Automatizar validación de integridad. | |

* **Historias técnicas 12**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-12 | **Título:** Logging, monitoreo y alertas |
| **Descripción:** | |
| Configurar sistema de registro centralizado de eventos, métricas de rendimiento y alertas proactivas ante errores o uso anómalo de recursos.  Aumenta la observabilidad del sistema y facilita la detección temprana de fallos. | |
| **Estimación:** 10 | **Dependiente de:** HT-03, HT-02 |
| **Prioridad: Media** |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Los logs se almacenan en formato JSON estructurado, con campos: timestamp, tipo, usuario, acción y severidad. * Alertas automáticas al correo/Slack cuando:   + CPU > 80 % durante 5 min.   + Espacio en disco < 15 %.   + Error 500 en más de 5 peticiones consecutivas. * Panel de monitoreo con métricas de disponibilidad y rendimiento (Prometheus/Grafana o similar). * En caso de caída de un servicio, alerta generada en < 60 segundos. * Los logs rotan automáticamente cada 7 días para evitar saturación. | |
| **Tareas:** | |
| * Configurar logging Odoo + Nginx. * Integrar Prometheus Node Exporter. * Crear dashboard básico. * Automatizar alertas por email. | |

* **Historias técnicas 13**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-13 | **Título:** Investigación comparativa: Python vs PHP en módulos Odoo |
| **Descripción:** | |
| Analizar técnicamente la viabilidad, rendimiento, mantenibilidad y seguridad del desarrollo de módulos en Python frente al uso de servicios externos en PHP.  Busca fundamentar la elección tecnológica más adecuada para el entorno Odoo del herbario. | |
| **Estimación:** 14 | **Dependiente de:** HT-01 |
| **Prioridad:** Media-Alta |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Informe comparativo con al menos cinco dimensiones: rendimiento, mantenimiento, comunidad, curva de aprendizaje y seguridad. * Prototipo funcional mínimo en Python (Odoo) y microservicio PHP conectado vía API REST. * Benchmark de latencia promedio ≤ 400 ms. * Identificación de riesgos y propuesta de mitigación. * Recomendación final documentada y validada con asesor técnico. * Si se detecta incompatibilidad crítica, registrar evidencia en informe. | |

* **Historias técnicas 14**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-14 | **Título:** Accesibilidad y experiencia de usuario (UX/UI responsive) |
| **Descripción:** | |
| Asegurar que la interfaz cumpla con principios de accesibilidad (WCAG 2.1) y se adapte a diferentes dispositivos (móvil, tablet, PC).  Mejora la usabilidad para usuarios con distintas capacidades y garantiza diseño inclusivo. | |
| **Estimación:** 8 | **Dependiente de:** HU-04 |
| **Prioridad:** Media |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Cumplimiento WCAG AA: contraste mínimo 4.5:1, navegación por teclado y etiquetas ARIA. * Diseño responsive hasta resolución mínima de 320 px. * Si un campo obligatorio no se completa, mostrar mensaje accesible por lector de pantalla. * El sistema debe mantener la coherencia visual en todos los navegadores modernos. * En caso de error de carga CSS, mostrar modo de accesibilidad simple (sin estilos). | |
| **Tareas:** | |
| * Revisar contraste y estructura semántica. * Ajustar CSS adaptable. * Validar compatibilidad móvil. | |

* **Historias técnicas 15**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-15 | **Título:** Procesos asíncronos y tareas en segundo plano (Jobs) |
| **Descripción:** | |
| Implementar ejecución de tareas pesadas en segundo plano mediante colas de trabajo (Odoo cron o Celery + Redis).  Esto evita bloquear la interfaz del usuario durante operaciones intensivas como generación de reportes o creación masiva de miniaturas. | |
| **Estimación:** 10 | **Dependiente de:** HT-03, HT-06 |
| **Prioridad:** Media |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * Las tareas asíncronas se encolan correctamente y se procesan sin bloquear la UI. * Tiempo de espera máximo permitido en cola: 10 minutos. * Si un job falla, se reintenta hasta 3 veces antes de generar alerta. * Se mantiene registro detallado de cada tarea (job\_log) con estado: pendiente, ejecutando, completado o error. * En caso de caída del servicio Celery, el sistema debe reanudar las tareas pendientes al reiniciar. | |
| **Tareas:** | |
| * Configurar broker (Redis). * Definir workers y jobs. * Implementar logs de ejecución. | |

* **Historias técnicas 16**

|  |  |
| --- | --- |
| **Anverso** | |
| **ID:** HT-16 | **Título: Pruebas de carga y escalabilidad (Locust)** |
| **Descripción:** | |
| Diseñar y ejecutar escenarios de carga que simulen usuarios concurrentes realizando operaciones críticas (búsquedas, subidas de imágenes, generación de reportes).  Determinar la capacidad máxima del sistema y puntos de saturación. | |
| **Estimación:** 10 | **Dependiente de:** HT-07, HT-03 |
| **Prioridad:** Media |
| **Reverso** | |
| **Criterios de aceptación:** | |
| * El sistema soporta al menos 100 usuarios concurrentes sin errores HTTP 500. * Tiempo medio de respuesta ≤ 3 s bajo carga normal. * Si se supera el umbral de rendimiento (CPU > 85 %), se genera alerta y se detiene la prueba. * Reporte generado con métricas: throughput, latencia promedio y desviación estándar. * En caso de fallo, registrar causa principal en performance\_report.log. | |
| **Tareas:** | |
| * Crear scripts Locust (búsqueda, CRUD, reportes). * Configurar umbrales de alerta. * Generar informe final. | |

**Product Backlog (Priorizado)**

|  |
| --- |
| HT-01 — Preparar entorno base: Odoo 17 (Infra). |
| HT-02 — PostgreSQL + backups (Infra). |
| HT-03 — Contenerización Docker (Infra). |
| HU-01 — Gestión de especímenes (CRUD + visual). |
| HT-05 — Seguridad y autenticación (OWASP). |
| HU-04 — Gestión de usuarios y permisos. |
| HU-09 — Historial de modificaciones |
| HT-08 — Framework de pruebas (unit/integration/e2e). |
| HU-03 — Búsqueda y filtros (optimización). |
| HT-07 — Optimización & índices. |
| HT-06 — Estrategia de almacenamiento de imágenes (decision + prototipo). |
| HU-02 — Gestión de imágenes y pipeline thumbnails. |
| HU-05 — Reportes PDF & exportaciones. |
| HT-15 — Asíncronía / Jobs (cola). |
| HU-06 — Estadísticas y visualizaciones. |
| HT-04 — Nginx + TLS. |
| HU-07 — Geolocalización y mapas. |
| HT-09 — Integración mapas (OSM vs Google) — decisión. |
| HT-11 — Política de backup y recuperación completa. |
| HT-12 — Logging / monitoring / alertas. |
| HT-13 — Investigación comparativa Python vs PHP (informe y prototipos). |
| HU-08 — QR y compartición. |
| HT-10 — QR service (implementación final). |
| HT-16 — Pruebas de carga & escalabilidad. |
| HT-14 — Accesibilidad & UX final polish. |

**Conclusiones**

La elaboración de las Historias de Usuario y Técnicas permitió estructurar de manera clara, ordenada y medible las funcionalidades y componentes tecnológicos del sistema HerbaProgram. Gracias a esta planificación, el proyecto cuenta con una pila de producto priorizada y completa, que abarca tanto los procesos funcionales del herbario (registro, búsqueda, control y reporte de especímenes) como los elementos técnicos esenciales para su correcto funcionamiento (configuración del entorno, seguridad, rendimiento, pruebas y mantenimiento).

Este enfoque integral garantiza que el desarrollo pueda ejecutarse de forma iterativa, minimizando riesgos y facilitando la adaptación ante cambios o mejoras. Además, deja sentadas las bases para la evolución futura del sistema, asegurando su escalabilidad, sostenibilidad y alineación con los objetivos de digitalización, preservación y difusión científica de la ESPOCH.