# Historias de Usuario Desglosadas en Tareas – Plataforma de Publicidad

## USU-01 – Registro de Usuario (con correo)

|  |  |
| --- | --- |
| Frontend | - Crear formulario de registro (correo, usuario, contraseña). - Validar correo y contraseña mínima de 8 caracteres, símbolo, mayúscula y minuscula. - Mostrar mensajes de error claros en español. |
| Backend | - Crear endpoint POST /users/register. - Validar correo único mediante repositorio JPA. - Encriptar contraseña antes de guardar con BCrypt. |
| QA/Pruebas | - Probar registro válido e inválido. - Verificar contraseñas encriptadas. |

## USU-02 – Registro con Google/Facebook

|  |  |
| --- | --- |
| Frontend | - Agregar botones de Google y Facebook en la UI de registro. - Gestionar redirección a login social. |
| Backend | - Configurar Spring Security con OAuth2 para Google y Facebook. - Validar que no se creen cuentas duplicadas. |
| QA/Pruebas | - Probar registro con Google y Facebook. - Validar vinculación de cuentas existentes. |

## USU-03 – Inicio de Sesión con correo

|  |  |
| --- | --- |
| Frontend | - Crear formulario de login (correo, contraseña). - Guardar token en cookie segura. |
| Backend | - Crear endpoint POST /users/login. - Validar credenciales con JPA. - Devolver JWT con tiempo de expiración. |
| QA/Pruebas | - Probar login válido e inválido. - Verificar expiración de token. |

## USU-04 – Inicio de Sesión con Google/Facebook

|  |  |
| --- | --- |
| Frontend | - Agregar opción de inicio con Google/Facebook en la UI. |
| Backend | - Validar token social y generar JWT propio. - Configurar integración OAuth2 con proveedores externos. |
| QA/Pruebas | - Probar inicio con Google/Facebook. - Revisar expiración de sesión. |

## USU-05 – Buscar prestadores por nombre

|  |  |
| --- | --- |
| Frontend | - Crear buscador en la página principal. - Mostrar resultados en cards responsivas. |
| Backend | - Crear endpoint GET /providers?name=xxx. - Implementar búsqueda con Spring Data JPA (LIKE). |
| QA/Pruebas | - Probar búsqueda con coincidencias y sin coincidencias. |

## USU-06 – Filtrar prestadores por rubro

|  |  |
| --- | --- |
| Frontend | - Agregar listado de rubros con checkboxs. - Permitir combinar filtro con búsqueda. |
| Backend | - Extender endpoint GET /providers con parámetro rubro. |
| QA/Pruebas | - Validar filtro individual y combinado con nombre. |

## USU-07 – Visualizar perfil de prestador

|  |  |
| --- | --- |
| Frontend | - Crear vista de detalle de prestador con foto y datos. - Mostrar calificación promedio y cantidad de opiniones. |
| Backend | - Crear endpoint GET /providers/{id}. |
| QA/Pruebas | - Verificar que los datos mostrados coincidan con el backend. |

## USU-08 – Calificar un prestador

|  |  |
| --- | --- |
| Frontend | - Agregar componente de estrellas clickeables en perfil. |
| Backend | - Crear endpoint POST /ratings. - Validar que un usuario solo pueda calificar una vez. |
| QA/Pruebas | - Probar calificación válida e intento de calificación duplicada. |

## USU-09 – Comentar sobre un prestador

|  |  |
| --- | --- |
| Frontend | - Agregar formulario para enviar comentarios en perfil. |
| Backend | - Crear endpoint POST /comments. - Validar autenticación para permitir comentarios. |
| QA/Pruebas | - Probar comentarios de usuarios logueados. - Verificar nombre de usuario y fecha en cada comentario. |

## ADM-01 – Alta de prestadores

|  |  |
| --- | --- |
| Frontend | - Crear formulario en el panel de administración. |
| Backend | - Crear endpoint POST /providers. - Validar rol ADMIN antes de registrar prestador. |
| QA/Pruebas | - Probar alta con pago validado y rechazar sin pago. |

## ADM-02 – Edición de prestadores

|  |  |
| --- | --- |
| Frontend | - Agregar formulario editable en panel de administración. |
| Backend | - Crear endpoint PUT /providers/{id}. - Mantener registro histórico de cambios. |
| QA/Pruebas | - Verificar cambios reflejados en frontend. |

## ADM-03 – Gestión de pagos adicionales

|  |  |
| --- | --- |
| Frontend | - Agregar campo para marcar pago validado en la edición. |
| Backend | - Validar pago antes de permitir cambios. - Registrar fecha y motivo de modificación. |
| QA/Pruebas | - Simular edición con y sin pago validado. |

## ADM-04 – Baja lógica de prestadores

|  |  |
| --- | --- |
| Frontend | - Agregar opción en panel para desactivar prestador. |
| Backend | - Crear endpoint PATCH /providers/{id}/deactivate. |
| QA/Pruebas | - Verificar que prestadores inactivos no aparezcan en búsquedas. |

## ADM-05 – Estadísticas del sistema

|  |  |
| --- | --- |
| Frontend | - Diseñar dashboard con gráficos simples en panel admin. |
| Backend | - Crear endpoint GET /admin/stats con métricas clave. |
| QA/Pruebas | - Validar precisión de las métricas mostradas. |

## RAN-01 – Orden por calificación

|  |  |
| --- | --- |
| Frontend | - Mostrar prestadores ordenados en frontend según rating. |
| Backend | - Calcular promedio de calificaciones en backend. |
| QA/Pruebas | - Verificar actualización tras nueva calificación. |

## RAN-02 – Orden alfabético en empate

|  |  |
| --- | --- |
| Frontend | - Mostrar resultados alfabéticamente si hay empate. |
| Backend | - Implementar orden secundario A-Z en backend. |
| QA/Pruebas | - Probar ranking con prestadores con misma calificación. |

## SEG-01 – Contraseñas encriptadas

|  |  |
| --- | --- |
| Frontend | - No requiere UI específica. |
| Backend | - Configurar Spring Security para encriptar con BCrypt. - Validar login comparando hash. |
| QA/Pruebas | - Revisar que contraseñas no estén en texto plano en DB. |

## SEG-02 – Restricción de comentarios y calificaciones

|  |  |
| --- | --- |
| Frontend | - Ocultar botones de calificación y comentarios si no hay sesión. |
| Backend | - Validar autenticación en endpoints de ratings y comments. |
| QA/Pruebas | - Probar intento de comentar/calificar sin sesión (redirigir a login). |

## SEG-03 – Acceso exclusivo al panel de administración

|  |  |
| --- | --- |
| Frontend | - No mostrar enlace al panel admin para usuarios sin rol ADMIN. |
| Backend | - Configurar middleware en Spring Security para validar rol ADMIN. |
| QA/Pruebas | - Probar acceso con y sin credenciales válidas (403 en no autorizados). |

## Rubros (Administrador)

### RUB-01 – Alta de Rubro

|  |  |
| --- | --- |
| Frontend | - Crear formulario en el panel de administración para registrar rubros. - Validar que el nombre del rubro sea único. |
| Backend | - Crear endpoint POST /rubros. - Validar nombre único mediante repositorio JPA. |
| QA/Pruebas | - Probar creación de rubro válido. - Rechazar intento de crear rubro duplicado. |

### RUB-04 – Listado de Rubros

|  |  |
| --- | --- |
| Frontend | - Crear vista de listado de rubros en panel de administración. - Agregar opciones de ordenar y filtrar por estado (activo/inactivo). |
| Backend | - Crear endpoint GET /rubros. - Incluir filtros por nombre y estado. |
| QA/Pruebas | - Verificar que se muestren todos los rubros con su estado. - Validar correcto funcionamiento de filtros y orden. |

### RUB-02 – Edición de Rubro

|  |  |
| --- | --- |
| Frontend | - Agregar formulario para editar rubros desde el listado. |
| Backend | - Crear endpoint PUT /rubros/{id}. - Validar que no se actualice a un nombre duplicado. |
| QA/Pruebas | - Probar edición de rubro con datos válidos. - Rechazar intento de edición con nombre duplicado. |

### RUB-03 – Baja Lógica de Rubro

|  |  |
| --- | --- |
| Frontend | - Agregar opción en listado para desactivar rubro. |
| Backend | - Crear endpoint PATCH /rubros/{id}/deactivate. - Mantener rubro en historial pero inactivo para nuevos registros. |
| QA/Pruebas | - Verificar que rubros inactivos no aparezcan como opción para prestadores nuevos. - Validar que prestadores con ese rubro no pierdan la categoría. |

### RUB-05 – Consulta de Rubro Específico

|  |  |
| --- | --- |
| Frontend | - Agregar opción para ver detalle de un rubro desde el listado. |
| Backend | - Crear endpoint GET /rubros/{id}. |
| QA/Pruebas | - Verificar que se muestren todos los datos del rubro. - Comprobar acceso desde listado. |