**Documento de**

**Épicas e Historias de Usuario**

**Vocational Insight**

***[VI]***

***Fecha: 20/09/2024***

**Tabla de contenido**

**Contenido**

[Datos del documento 3](#_heading=h.1fob9te)

[Épicas para el proyecto Vocational Insight 4](#_heading=h.tyjcwt)

[Priorización de Épicas 4](#_heading=h.3dy6vkm)

[Definición de Historias de Usuario 5](#_heading=h.1t3h5sf)

# Datos del documento

Histórico de Revisiones

| Versión | Fecha | Descripción/cambio | autor |
| --- | --- | --- | --- |
| 1.0 | 20/09/2024 | Épicas agregadas y priorización | Equipo completo. |
| 2.0 | 21/09/2024 | Definición de historias de usuario | Equipo completo. |

Información del Proyecto

| Organización | Duoc UC. Escuela de Informática y Telecomunicaciones |
| --- | --- |
| Sección | 003V |
| Proyecto (Nombre) | Vocational Insight |
| Fecha de Inicio | 14/08/2024 |
| Fecha de Término | 07/12/2024 |
| Patrocinador principal | Duoc UC |
| Docente | Jorge Gomez |

Integrantes

| Rut | Nombre | Correo |
| --- | --- | --- |
| 19152531-0 | Kristel Mercurino | k.mercurino@duocuc.cl |
| 16.971.388-K | Daniel Canales | da.canales@duocuc.cl |
| 21.987.234-0 | Luciano Elgueda | luc.elgueda@duoccuc.cl |

# Épicas para el proyecto Vocational Insight

| **ID** | **Épica** |
| --- | --- |
| 1 | Implementar el sistema de registro y autenticación de usuarios |
| 2 | Implementar el test vocacional personalizado |
| 3 | Desarrollar el sistema de recomendaciones de carrera |
| 4 | Ver historial de resultados |
| 5 | Implementar la funcionalidad de análisis de datos de empleabilidad y salarios |
| 6 | Crear el comparador de carreras |
| 7 | Desarrollar el sistema de exportación de informes en PDF |
| 8 | Integrar un boletín informativo personalizado |
| 9 | Implementar la seguridad y privacidad de los datos de los usuarios |
| 10 | Desarrollar el sistema de feedback de usuarios |

# Priorización de Épicas

| Para la estimación de nuestras épicas en el proyecto *Vocational Insight* utilizamos la técnica de MoSCoW debido a su simplicidad y eficacia, incluso en la etapa de desarrollo en la que nos encontramos. Actualmente, estamos en pleno desarrollo del código, por lo que ya tenemos una visión clara de las tareas y los objetivos que debemos cumplir. Elegimos MoSCoW porque nos permitió clasificar las épicas de manera rápida y eficiente, basándonos en la complejidad y el esfuerzo necesario para cada una sin entrar en detalles técnicos excesivos.  Esta técnica nos facilitó la discusión en equipo y el consenso sobre la magnitud de cada tarea. Al asignar las categorías:   * **Debe tener (Must Have)**: es crítico y prioritario. * **Debería tener (Should Have)**: es importante, pero no tan urgente como los "Debe tener". * **Podría tener (Could Have)**: es deseable y útil, pero no urgente.   A las épicas, logramos mantener una evaluación ágil de las tareas acorde al ritmo de desarrollo en el que estamos inmersos. Aunque estamos en una etapa avanzada del proyecto, la clasificación mediante MoSCoW sigue siendo útil para identificar rápidamente qué tareas requieren más recursos y tiempo.  El uso de esta técnica también nos permitió equilibrar la carga de trabajo entre los miembros del equipo, asignando de manera eficiente las épicas más grandes o complejas a aquellos con mayor capacidad o tiempo disponible, sin necesidad de complicadas estimaciones numéricas. Esto ha sido especialmente valioso para mantener el flujo de trabajo a buen ritmo y evitar cuellos de botella.  En resumen, el uso de MoSCoW fue conveniente porque:   * Aun en esta fase avanzada de desarrollo, nos permitió priorizar las épicas de forma clara y eficiente. * Facilitó la participación de todo el equipo en la discusión sobre el esfuerzo necesario para cada tarea, lo que ayudó a alinear expectativas. * Nos permitió balancear la carga de trabajo de manera efectiva y mantener un flujo de desarrollo ágil. |
| --- |

| **ID** | **Épica** | **Priorización** |
| --- | --- | --- |
| 1 | Implementar el sistema de registro y autenticación de usuarios | Debe tener |
| 2 | Implementar el test vocacional personalizado | Debe tener |
| 3 | Desarrollar el sistema de recomendaciones de carrera | Debe tener |
| 4 | Ver historial de resultados | Debería tener |
| 5 | Implementar la funcionalidad de análisis de datos de empleabilidad y salarios | Podría tener |
| 6 | Crear el comparador de carreras | Debe tener |
| 7 | Desarrollar el sistema de exportación de informes en PDF | Podría tener |
| 8 | Integrar un boletín informativo personalizado | Podría tener |
| 9 | Implementar la seguridad y privacidad de los datos de los usuarios | Debe tener |
| 10 | Desarrollar el sistema de feedback de usuarios | Podría tener |

### **Explicación de la Priorización**

**Implementar el sistema de registro y autenticación de usuarios  
Debe tener (Must Have)**: Esta funcionalidad es fundamental para que los usuarios puedan acceder al sistema. Sin un sistema de registro y autenticación, los usuarios no podrán interactuar con la plataforma, lo que hace que sea un componente crítico para el proyecto.

**Implementar el test vocacional personalizado  
Debe tener (Must Have)**: El test vocacional personalizado es esencial para la funcionalidad principal del sistema, ya que permite a los usuarios recibir recomendaciones relevantes basadas en sus intereses y habilidades. Sin esta funcionalidad, el sistema no cumpliría su propósito.

**Desarrollar el sistema de recomendaciones de carrera  
Debe tener (Must Have)**: La recomendación de carreras es el núcleo del proyecto. Este sistema proporciona el valor principal al usuario, y sin él, la plataforma perdería su razón de ser. Dada su complejidad y relevancia, es una prioridad absoluta.

**Ver historial de resultados  
Debería tener (Should Have)**: Esta funcionalidad es importante para la experiencia del usuario, ya que permite visualizar las recomendaciones y decisiones pasadas. Sin embargo, el sistema podría funcionar inicialmente sin esta característica, lo que la convierte en una prioridad alta pero no crítica.

**Implementar la funcionalidad de análisis de datos de empleabilidad y salarios  
Podría tener (Could Have)**: Aunque útil para enriquecer la experiencia del usuario, esta funcionalidad es más avanzada y no es esencial en las primeras etapas del sistema. Si hay tiempo y recursos adicionales, puede implementarse más adelante.

**Crear el comparador de carreras  
Debe tener (Must Have)**: Esta herramienta es crucial para que los usuarios puedan tomar decisiones informadas al comparar diferentes opciones de carrera. Al igual que el sistema de recomendaciones, esta funcionalidad es central para el propósito del sistema.

**Desarrollar el sistema de exportación de informes en PDF  
Podría tener (Could Have)**: La capacidad de exportar informes en PDF es un complemento útil, pero no esencial para el funcionamiento básico del sistema. Puede ser implementado en una fase posterior si el tiempo lo permite.

**Integrar un boletín informativo personalizado  
Podría tener (Could Have)**: Un boletín personalizado mejora la experiencia del usuario, pero su ausencia no afecta el funcionamiento principal del sistema. Es una funcionalidad de valor añadido que puede desarrollarse más adelante.

**Implementar la seguridad y privacidad de los datos de los usuarios  
Debe tener (Must Have)**: Dado que el sistema maneja información sensible de los usuarios, es crucial garantizar la seguridad y privacidad de los datos desde el principio. Sin esta funcionalidad, el sistema sería vulnerable y podría perder la confianza de los usuarios.

**Desarrollar el sistema de feedback de usuarios  
Podría tener (Could Have)**: El sistema de feedback es una herramienta útil para mejorar la plataforma a lo largo del tiempo, pero no es esencial en la primera fase. Su desarrollo puede postergarse hasta que las funcionalidades más críticas estén implementadas.

### **Resumen de la Priorización:**

**Debe tener (Must Have)**: Estas funcionalidades son esenciales para que el sistema sea utilizable y cumpla su objetivo principal (Registro y autenticación, test vocacional, recomendaciones de carrera, comparador de carreras, y seguridad de datos).

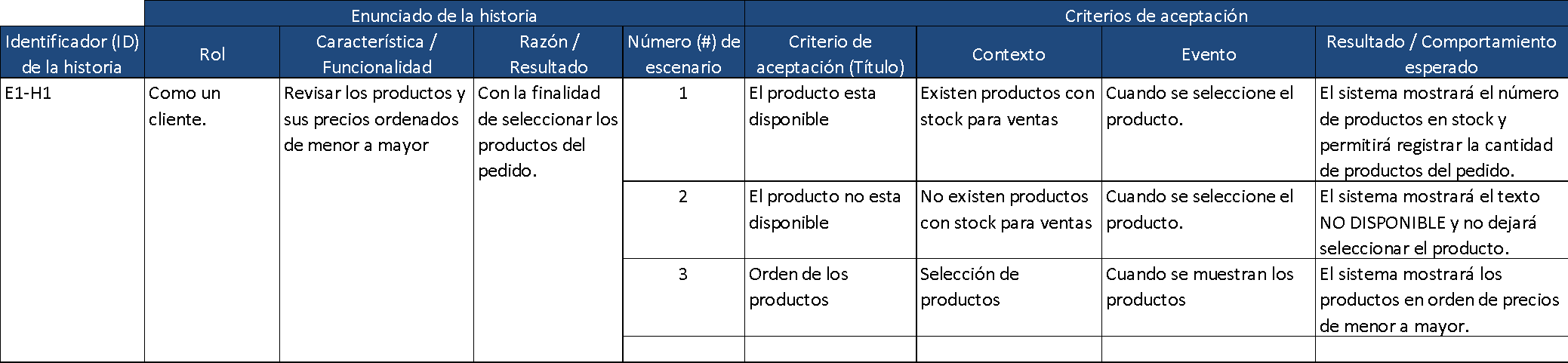
**Debería tener (Should Have)**: Aunque importantes, las funcionalidades en esta categoría (historial de resultados) pueden implementarse después del lanzamiento inicial.

**Podría tener (Could Have)**: Características como análisis de empleabilidad, exportación de informes, boletín informativo y sistema de feedback son deseables pero no esenciales para el funcionamiento básico del sistema en su fase inicial.

# Definición de Historias de Usuario

[2.2 Plantilla Historias Usuario - Oficina de Proyectos de Informatica.xlsx](https://docs.google.com/spreadsheets/d/1Og4UDyybDZF2T1WP-nloRB0un0BNIO4I/edit?usp=drive_link&ouid=100502663691709774700&rtpof=true&sd=true)

Utilice el ejemplo y la plantilla en Excel proporcionada por el docente, copie la información e insértela como imagen de acuerdo al siguiente ejemplo:



Historia 1

Historia 2

Historia 3