#### **Documento de Pruebas de Software**

#### **Proyecto de Rector Simulator**

#### **Pruebas de Base de Datos**

**Prueba 1: Verificación de conexión a Base de datos.**

| **ID Caso** | PBD-001 |
| --- | --- |
| **Descripción** | Comprobación de que la conexión a la base de datos se establece correctamente |
| **Objetivo** | Validar que la base de datos está activa y acepta conexiones. |
| **Consulta SQL** | SELECT 1; |
| **Resultado Esperado** | La consulta devuelve el resultado 1, indicando que la conexión es exitosa. |
| **Resultado Obtenido** | La consulta devolvió el resultado 1. |
| **Estado** | Exitoso |

**Prueba 2: Validación de restricción NOT NULL en campos de tabla Loggin\_jugador**

| **ID Caso** | PBD-002 |
| --- | --- |
| **Descripción** | Comprobación de que los campos de la tabla Loggin\_jugador que están configurados como NOT NULL no permiten valores vacíos o nulos. |
| **Objetivo** | Validar que los campos clave de la tabla Loggin\_jugador (e.g., email, username, password) no permitan valores nulos, garantizando la integridad de los datos. |
| **Consulta SQL** | sql INSERT INTO Loggin\_jugador (usuario, email, username, password, pais, date\_joined) VALUES ('test\_user', NULL, 'test\_username', 'test\_password', 'Peru', NOW()); |
| **Resultado Esperado** | La consulta genera un error indicando que el campo email no puede ser nulo. |
| **Resultado Obtenido** | La base de datos devolvió un error de restricción NOT NULL para el campo de email. |
| **Estado** | Exitoso |

**Prueba 3: Validación de Restricción de Unicidad en los Campos de la Tabla Loggin\_jugador**

| **ID Caso** | PBD-003 |
| --- | --- |
| **Descripción** | Verificar que los campos configurados como únicos en la tabla Loggin\_jugador no permitan valores duplicados. |
| **Objetivo** | Asegurarse de que las restricciones de unicidad en la tabla son efectivas para garantizar la integridad de los datos. |
| **Consulta SQL** | sql INSERT INTO Loggin\_jugador (usuario, email, username, password, pais, date\_joined) VALUES ('test\_user', 'test\_email@example.com', 'test\_username', 'password1', 'Peru', NOW());  INSERT INTO Loggin\_jugador (usuario, email, username, password, pais, date\_joined) VALUES ('test\_user2', 'test\_email@example.com', 'test\_username', 'password2', 'Peru', NOW()); |
| **Resultado Esperado** | La segunda consulta genera un error debido a la duplicidad de los valores en las columnas con restricciones únicas (email y username). |
| **Resultado Obtenido** | Error generado |
| **Estado** | Exitoso |

#### **Pruebas de Requisitos funcionales**

**Prueba 1: Validación de Sistema de Toma de Decisiones**

| **ID Caso** | PRF-001 |
| --- | --- |
| **Descripción** | Verificar que el jugador puede tomar decisiones basadas en eventos aleatorios mientras no pierda la partida. |
| **Objetivo** | Validar que en cada evento aleatorio el jugador pueda tomar decisiones entre 2 opciones sin problema alguno. |
| **Procedimiento** | 1. Jugar una partida  2. Se presenta una situación aleatoria.  3. Tomar una decisión  4. Esperar comportamiento del sistema |
| **Resultado Esperado** | El jugador puede tomar decisiones que le presenta el sistema mientras este no pierda la partida. |
| **Resultado Obtenido** | Las tomas de decisiones en diferentes eventos no tuvieron complicación alguna. |
| **Estado** | Exitoso |

**Prueba 2: Validación de Gestión de Aprobación del Alumnado**

| **ID Caso** | PRF-002 |
| --- | --- |
| **Descripción** | Verificar que el jugador puede observar y gestionar la aprobación del alumnado a través de una barra de progreso actualizada en tiempo real y gestionarla por medio de decisiones. |
| **Objetivo** | Validar que la barra de progreso refleja los cambios en la aprobación estudiantil según las decisiones tomadas. |
| **Procedimiento** | 1. Jugar una partida.  2. Tomar decisiones que afecten la aprobación estudiantil.  3. Observar la barra de progreso en tiempo real. |
| **Resultado Esperado** | La barra de progreso se actualiza inmediatamente tras cada decisión que impacta la aprobación estudiantil. |
| **Resultado Obtenido** | La barra de progreso reaccionó correctamente a los cambios de aprobación. |
| **Estado** | Exitoso |

**Prueba 3: Validación de Gestión de Recursos Financieros**

| **ID Caso** | PRF-003 |
| --- | --- |
| **Descripción** | Verificar que el jugador puede ver y asignar fondos dentro de la universidad y que estas acciones impacten en su estabilidad financiera. |
| **Objetivo** | Validar que las decisiones financieras afecten la estabilidad financiera según lo esperado. |
| **Procedimiento** | 1. Jugar una partida.  2.Ver estado de los recursos.  3. Asignar fondos a distintas áreas cuando se tenga oportunidad o no asignar.  4. Verificar los cambios generados en los recursos. |
| **Resultado Esperado** | La estabilidad financiera se actualiza correctamente tras cada reasignación |
| **Resultado Obtenido** | Los cambios financieros impactaron en la estabilidad financiera como se esperaba. |
| **Estado** | Exitoso |

**Prueba 4: Validación de Gestión de Recursos Financieros**

| **ID Caso** | PRF-004 |
| --- | --- |
| **Descripción** | Verificar que el sistema guarda correctamente los puntajes del jugador y actualiza el ranking según el desempeño. |
| **Objetivo** | Validar la persistencia de puntajes y su comparación con los registros previos para actualizar el ranking. |
| **Procedimiento** | 1. Jugar una partida y finalizar con un puntaje.  2. Verificar que el puntaje se almacena en la base de datos.  3. Comparar el puntaje actual con los anteriores y observar el ranking actualizado. |
| **Resultado Esperado** | El puntaje se guarda correctamente y el ranking se actualiza según el desempeño. |
| **Resultado Obtenido** | El sistema registró el puntaje y actualizó el ranking correctamente. |
| **Estado** | Exitoso |

**Prueba 5: Validación de Eventos Aleatorios**

| **ID Caso** | PRF-005 |
| --- | --- |
| **Descripción** | Verificar que el sistema genera eventos aleatorios durante la partida y que estos afectan directamente la jugabilidad. |
| **Objetivo** | Validar que los eventos aleatorios tienen un impacto en las estadísticas del jugador. |
| **Procedimiento** | 1. Jugar una partida.  2. Verificar la generación de eventos aleatorios.  3. Tomar decisiones basadas en los eventos presentados y observar los impactos. |
| **Resultado Esperado** | Se generan eventos aleatorios variados y sus consecuencias afectan las estadísticas del jugador. |
| **Resultado Obtenido** | Los eventos aleatorios se generaron y afectaron las estadísticas correctamente. |
| **Estado** | Exitoso |

**Prueba 6: Validación de Múltiples Finales**

| **ID Caso** | PRF-006 |
| --- | --- |
| **Descripción** | Verificar que el juego ofrece múltiples finales según las decisiones tomadas por el jugador a lo largo de la partida. |
| **Objetivo** | Validar que las decisiones del jugador determinan el final alcanzado en la partida. |
| **Procedimiento** | 1. Jugar varias partidas tomando decisiones diferentes.  2. Observar los finales generados en función de las decisiones. |
| **Resultado Esperado** | El juego genera distintos finales basados en las decisiones del jugador. |
| **Resultado Obtenido** | Los finales variaron según las decisiones tomadas. |
| **Estado** | Exitoso |