



TASK HARMONY

# Trabajo de Fin de Grado.

<b>Primera fase.</b>	<b>4</b>
- Empresa Cliente:	4
Home Dynamics Inc.	4
- Funcionalidades:	5
Gestión de Usuarios y Familias:	5
Manejo de Tareas:	6
Sistema de Premios:	6
Notificaciones:	6
Interfaz de Usuario:	6
- Tecnologías usadas:	6
Java:	6
JPA (Java Persistence API):	6
Hibernate:	7
Swing:	7
Maven:	7
MySQL/PostgreSQL/H2:	7
- Relación con las asignaturas del módulo:	7
<b>Segunda fase.</b>	<b>8</b>
- Pruebas	8
Pruebas de caja negra	8
Identificación del Caso de Prueba: CP001	8
Pruebas de caja blanca	10
Identificación del Caso de Prueba: CP002	10
Pruebas unitarias	11
Identificación del Caso de Prueba: CP003	11
Pruebas de integración	13
Identificación del Caso de Prueba: CP004	13
Pruebas Funcionales	14
Identificación del Caso de Prueba: CP005	14
<b>Tercera fase.</b>	<b>16</b>
- Manual de usuario	16
- Documento descriptivo de la base de datos	16
Diseño Relacional	16
Modelo E-R	17
Entidades y Atributos	18
Relaciones	20

2ºC.F.G.S. Desarrollo de Aplicaciones Multiplataforma. IES Padre Suárez.

Martín Mrowczynski-Van Allen Romero.

Atributos de Relación	20
Normalización	21
Primera Forma Normal (1FN)	21
Segunda Forma Normal (2FN)	21
Tercera Forma Normal (3FN)	22
- Guía de instalación	22

## Primera fase.

- Empresa Cliente:

Home Dynamics Inc.



**Home Dynamics Inc.** es una startup tecnológica enfocada en el desarrollo y distribución de soluciones digitales para mejorar la gestión y dinámica de los hogares modernos. La empresa identifica una necesidad creciente en el mercado de productos que no solo optimizan las tareas domésticas, sino que también fomentan un ambiente familiar colaborativo y motivador.

### **Misión:**

2º C.F.G.S. Desarrollo de Aplicaciones Multiplataforma. IES Padre Suárez.  
Martín Mrowczynski-Van Allen Romero.

Impulsar la armonía y eficiencia en los hogares a través de soluciones tecnológicas innovadoras que promueven la colaboración, la responsabilidad y la recompensa entre sus miembros.

#### **Visión:**

Ser líderes en la creación de entornos familiares donde la tecnología y la gamificación conviertan la gestión del hogar en una experiencia gratificante y efectiva.

En el mercado actual, las **familias modernas** enfrentan el desafío de equilibrar el **trabajo**, la **educación** y el tiempo de **ocio**, lo que crea una **necesidad** evidente de herramientas efectivas para la **gestión de tareas domésticas** que distribuyan equitativamente las responsabilidades, mejorando así la organización del hogar. Además, hay una **creciente demanda** por sistemas que no solo asignen tareas, sino que también **incentiven a los miembros del hogar**, en particular a niños y adolescentes, **a cumplir** con estas mediante recompensas tangibles o **experiencias gratificantes**. Estos productos deben ser **sencillos y accesibles** para todos los miembros de la familia, independientemente de su habilidad tecnológica, para asegurar una **adopción** amplia y **efectiva**. Asimismo, con el aumento de **hogares inteligentes**, surge una **oportunidad** única para integrar aplicaciones de gestión de tareas con otros dispositivos del hogar, como asistentes de voz y smartphones, maximizando su utilidad y conveniencia en un futuro.

Home Dynamics Inc. podría usar "**Task Harmony**" para **satisfacer las necesidades** específicas de gestión de tareas en el hogar, proporcionando una **solución** tanto práctica como atractiva para las familias modernas que buscan **optimizar su tiempo y mejorar la colaboración** doméstica. Dada la creciente demanda de soluciones de gestión de tareas domésticas, existe un amplio mercado potencial. La empresa también podría explorar **oportunidades de personalización y venta cruzada**, adaptando "Task Harmony" a diferentes tipos de hogares y necesidades familiares, y **potencialmente integrándolo con otros productos** o servicios para el hogar. Además, el modelo inicialmente enfocado en familias tiene el **potencial de expandirse para incluir otros tipos de comunidades**, como residencias compartidas de estudiantes o grupos de convivencia senior, lo que permite una escalabilidad considerable de la plataforma para adaptarse a diversas necesidades.

- Funcionalidades:

## **Gestión de Usuarios y Familias:**

Los usuarios pueden registrarse, iniciar sesión y cerrar sesión. Cada usuario puede pertenecer a una o más familias, pero solo pueden administrar una y una familia solo puede ser administrada por un usuario. Los administradores de las familias pueden aceptar o rechazar solicitudes de nuevos miembros.

2ºC.F.G.S. Desarrollo de Aplicaciones Multiplataforma. IES Padre Suárez.

Martín Mrowczynski-Van Allen Romero.

## Manejo de Tareas:

Los administradores pueden crear, editar y activar tareas para los miembros de la familia. Las tareas tienen descripciones, puntos asignados (basados en la dificultad o importancia) y estado (pudiendo ser activa o inactiva). Los usuarios registran las tareas que han completado, y estas acciones son verificadas por los administradores, quienes asignan la calificación de la tarea a los usuarios.

## Sistema de Premios:

Los puntos obtenidos por completar tareas pueden ser canjeados por premios. Los premios son definidos por el administrador de la familia y pueden incluir cosas como permisos especiales, pequeñas recompensas económicas, o actividades recreativas. Los usuarios pueden ver los premios disponibles y reclamarlos según los puntos que hayan acumulado.

## Notificaciones:

El sistema envía notificaciones para informar a los administradores sobre nuevas solicitudes de membresía en la familia.

## Interfaz de Usuario:

La aplicación cuenta con una interfaz gráfica amigable que permite a los usuarios navegar fácilmente a través de las diferentes secciones como tareas, premios, y configuraciones de la familia gracias a un menú lateral que permite el acceso rápido a las principales

- Tecnologías usadas:

## Java:

Lenguaje de programación principal utilizado para el desarrollo del backend y la interfaz gráfica (GUI). He basado la mayor parte de mi proyecto en este lenguaje por ser al que más tiempo le he dedicado y mejor manejo, además de haber realizado prácticas relacionadas con la funcionalidad de mi programa anteriormente.

## JPA (Java Persistence API):

Utilizado para gestionar la persistencia de datos en la base de datos. Como he mencionado anteriormente, he realizado otras actividades que requerían de acceso a una base

de datos desde java y esta herramienta ejerce una función clave en este aspecto siendo además fácil de aprender y cubrir plenamente mis necesidades.

## Hibernate:

Es una implementación de JPA y se utiliza para el mapeo objeto-relacional, facilitando la integración entre Java y sistemas de base de datos relacional sin necesidad de escribir consultas SQL complejas directamente.

## Swing:

Usado para construir la interfaz gráfica de usuario en la aplicación de escritorio. Al basar mi programa en Java, he optado por desarrollar la interfaz en el mismo lenguaje, apoyándome en la experiencia adquirida en tareas anteriores, en la asignatura de "Desarrollo de interfaces".

## Maven:

Herramienta de gestión y construcción de proyectos que también maneja dependencias. La razón de su uso es la misma que para JPA.

## MySQL/PostgreSQL/H2:

Como sistema de gestión de bases de datos relacionales para el almacenamiento de datos. Se han convertido en unas herramientas fundamentales a la hora de tratar con bases de datos.

- Relación con las asignaturas del módulo:

Asignatura	Relación
Programación	El proyecto hace uso extenso de programación en Java para desarrollar la aplicación y la lógica de negocio.
Bases de Datos	El proyecto utiliza bases de datos para almacenar y gestionar los datos de usuarios, tareas, premios, etc., utilizando JPA para la persistencia de datos.
Sistemas informáticos	El proyecto puede implicar configuración de entornos de desarrollo y pruebas en

	diferentes sistemas operativos.
<b>Entornos de desarrollo</b>	El proyecto se desarrolla utilizando Maven y entornos de desarrollo integrados (IDEs).
<b>Acceso a datos</b>	El proyecto emplea JPA como parte del acceso a datos para interactuar con la base de datos de forma más eficiente y abstracta.
<b>Desarrollo de interfaces</b>	El proyecto utiliza Java Swing para desarrollar la interfaz gráfica, permitiendo una interacción amigable con el usuario.
<b>Empresa e iniciativa emprendedora</b>	El proyecto crea/identifica una necesidad, la ausencia de un sistema gestor de tareas domésticas, que suma con el aumento de la demanda de apps y programas de uso diario, pudiendo presentar el proyecto como idea de producto.

## Segunda fase.

- Pruebas

### Pruebas de caja negra

#### Identificación del Caso de Prueba: CP001

**Título del Caso de Prueba:** Verificación de Inicio de Sesión

**Descripción:** Este caso de prueba tiene como objetivo verificar que el sistema de inicio de sesión funcione correctamente, asegurando que solo los usuarios con credenciales válidas puedan acceder al sistema.

**Requisitos Asociados:**

- **R001:** El sistema debe permitir el inicio de sesión utilizando un DNI y contraseña.

2º C.F.G.S. Desarrollo de Aplicaciones Multiplataforma. IES Padre Suárez.  
Martín Mrowczynski-Van Allen Romero.



- **R002:** El sistema debe rechazar el acceso a usuarios con credenciales inválidas.

#### **Pasos de Ejecución:**

1. Abrir la aplicación y dirigirse a la pantalla de inicio de sesión.
2. Ingresar un DNI y contraseña válidos en los campos correspondientes.
3. Hacer clic en el botón de "Login".
4. Repetir los pasos utilizando credenciales inválidas.

#### **Datos de Entrada:**

- Entrada válida: DNI = "12345678A", Contraseña = "password123"
- Entrada inválida: DNI = "87654321B", Contraseña = "incorrect"

#### **Condiciones Previas:**

- La aplicación debe estar instalada y operativa.
- Los usuarios de prueba deben estar previamente registrados en la base de datos.

#### **Resultado Esperado:**

- Con credenciales válidas: El usuario debe ser redirigido a la página de selección de familias.
- Con credenciales inválidas: Debe aparecer un mensaje de error indicando que el DNI o la contraseña son incorrectos.

#### **Criterios de Aceptación:**

- El acceso solo se concede con credenciales válidas.
- Se muestra un mensaje de error claro y específico con credenciales inválidas.

#### **Entorno de Prueba:**

- Sistema operativo: Windows 10
- Versión de la aplicación: 1.0.0
- Conexión a base de datos activa y accesible

#### **Notas Adicionales:**

- Asegurarse de probar con varios usuarios para verificar la robustez del control de acceso.
- Considerar la sensibilidad a mayúsculas y minúsculas en las contraseñas.

#### **Estado del Caso de Prueba:** Éxito

# Pruebas de caja blanca

## Identificación del Caso de Prueba: CP002

**Título del Caso de Prueba:** Prueba de Cobertura de Código para la Funcionalidad de Registro

**Descripción:** Este caso de prueba se enfoca en evaluar la cobertura de código y los caminos de ejecución internos del sistema al realizar el registro de usuarios, asegurando que todos los posibles caminos de ejecución sean probados y validados.

### Requisitos Asociados:

- **R003:** El sistema debe permitir el registro de nuevos usuarios con DNI, nombre, apellidos y contraseña.
- **R004:** El sistema debe validar la unicidad del DNI antes de permitir el registro.

### Pasos de Ejecución:

1. Abrir el módulo de pruebas unitarias en el entorno de desarrollo.
2. Ejecutar pruebas unitarias que cubran todos los métodos y funciones relacionadas con el registro de usuarios.
3. Verificar que cada condición, bucle y rama en el código sean ejecutados al menos una vez.
4. Documentar cualquier desviación o error encontrado durante la prueba.

### Datos de Entrada:

- DNI: "12345678A", "12345678A" (**duplicado para probar la validación**), "98765432B"
- Nombre: "Juan", "Ana"
- Apellidos: "Pérez", "Gómez"
- Contraseña: "pass123", "abc123"

### Condiciones Previas:

- Acceso al código fuente y al entorno de desarrollo.
- Uso del "debug" para realizar un seguimiento del proceso.

### Resultado Esperado:

- Todos los caminos de ejecución relacionados con la validación de DNI duplicados y el registro exitoso deben ser probados.
- No deben haber errores no capturados o excepciones durante la ejecución de las pruebas.

**Criterios de Aceptación:**

- Cobertura de código completa sobre las áreas críticas.
- Manejo adecuado de DNI duplicados.
- Registro exitoso de usuarios no duplicados.

**Entorno de Prueba:**

- IDE: Netbeans
- Lenguaje de programación: Java.
- Versión de la aplicación: 1.0.0
- Herramienta de cobertura de código: Debug.

**Notas Adicionales:**

- Prestar atención a los posibles errores de entrada y asegurarse de que el sistema los maneje adecuadamente.
- Considerar pruebas adicionales para casos extremos no cubiertos inicialmente.

**Estado del Caso de Prueba:** **Éxito**

## Pruebas unitarias

### Identificación del Caso de Prueba: CP003

**Título del Caso de Prueba:** Pruebas Unitarias para la Función de Registro de Usuario

**Descripción:** Este caso de prueba verifica que la función de registro de usuarios en el sistema funcione correctamente, evaluando cada componente individual (métodos y funciones) que participa en el proceso de registro.

**Requisitos Asociados:**

- **R003:** El sistema debe permitir el registro de nuevos usuarios con DNI, nombre, apellidos y contraseña.
- **R004:** El sistema debe validar la unicidad del DNI antes de permitir el registro.

**Pasos de Ejecución:**

1. Crear un conjunto de pruebas que incluya:
  - Prueba con datos válidos para el registro.
  - Prueba para verificar el manejo de DNI duplicados.
  - Pruebas para verificar la validación de campos vacíos o incorrectos.
2. Ejecutar las pruebas unitarias.

2º C.F.G.S. Desarrollo de Aplicaciones Multiplataforma. IES Padre Suárez.

Martín Mrowczynski-Van Allen Romero.

3. Verificar los resultados de cada prueba contra los resultados esperados.

**Datos de Entrada:**

- **Caso 1:** DNI: "12345678A", Nombre: "Juan", Apellidos: "Pérez", Contraseña: "pass123"
- **Caso 2:** DNI: "12345678A" (**duplicado**), Nombre: "Ana", Apellidos: "Gómez", Contraseña: "abc123"
- **Caso 3:** DNI: "", Nombre: "Prueba", Apellidos: "**Vacío**", Contraseña: "test123"

**Condiciones Previas:**

- La base de datos de prueba debe estar configurada y accesible.

**Resultado Esperado:**

- Caso 1: Debe completar el registro y devolver un mensaje de éxito.
- Caso 2: Debe fallar debido a DNI duplicado y devolver un mensaje de error.
- Caso 3: Debe fallar por datos incompletos y devolver un mensaje de error.

**Criterios de Aceptación:**

- Todas las pruebas deben pasar conforme a los resultados esperados.
- Los mensajes de error deben ser claros y específicos para cada tipo de fallo.

**Entorno de Prueba:**

- Sistema operativo: Windows 10.
- IDE: Netbeans.
- Base de datos: base de datos de prueba.

**Notas Adicionales:**

- Asegúrese de que las pruebas no afecten la base de datos de producción.
- Revise los logs para cualquier error inesperado o advertencias durante las pruebas.

**Estado del Caso de Prueba:**

- **Caso 1:** **Éxito**
- **Caso 2:** **Éxito**
- **Caso 3:** **Éxito**

# Pruebas de integración

## Identificación del Caso de Prueba: CP004

**Título del Caso de Prueba:** Pruebas de Integración para el Sistema de Registro y Autenticación

**Descripción:** Este caso de prueba verifica la correcta integración y cooperación entre los módulos de registro de usuarios y autenticación, asegurando que los datos registrados puedan ser utilizados inmediatamente para la autenticación en el sistema.

**Requisitos Asociados:**

- **R003:** El sistema debe permitir el registro de nuevos usuarios con DNI, nombre, apellidos y contraseña.
- **R005:** El sistema debe permitir que los usuarios registrados se autenticquen usando su DNI y contraseña.

**Pasos de Ejecución:**

1. Crear un caso de prueba que incluya:
  - Registro de un usuario nuevo con datos válidos.
  - Inmediatamente después del registro, intentar autenticar al usuario con los mismos datos.
2. Ejecutar la prueba y observar la interacción entre los módulos de registro y autenticación.
3. Verificar que la autenticación sea exitosa utilizando los datos recién registrados.

**Datos de Entrada:**

- DNI: "87654321B", Nombre: "Laura", Apellidos: "Martínez", Contraseña: "securePass123"

**Condiciones Previas:**

- La aplicación debe estar configurada para conectarse a una base de datos de prueba.
- Todos los servicios y controladores involucrados en el registro y autenticación deben estar operativos.

**Resultado Esperado:**

- El usuario debe ser capaz de registrarse y luego autenticarse exitosamente sin errores, demostrando la correcta integración entre los módulos.

**Criterios de Aceptación:**

- El caso de prueba se considera exitoso si el usuario puede autenticarse inmediatamente después de su registro usando los datos proporcionados.
- No deben surgir errores de integración durante el proceso.

#### **Entorno de Prueba:**

- IDE: NetBeans.
- Sistema Operativo: Windows 10.
- Base de datos: base de datos de prueba.

#### **Notas Adicionales:**

- Es esencial que las transacciones de base de datos se manejen correctamente para evitar estados inconsistentes entre el registro y la autenticación.
- La configuración del entorno de prueba debe garantizar que no hay datos residuales que puedan afectar los resultados de las pruebas, por ello se usa una base de datos limpia.

**Estado del Caso de Prueba:** **Éxito**

## **Pruebas Funcionales**

### **Identificación del Caso de Prueba: CP005**

**Título del Caso de Prueba:** Pruebas Funcionales para la Gestión de Tareas

**Descripción:** Este caso de prueba evalúa el comportamiento funcional del sistema de gestión de tareas, verificando que el administrador pueda crear, editar, y visualizar tareas de acuerdo a los requisitos funcionales establecidos. Este proceso garantiza que las funciones estén alineadas con las expectativas del administrador y cumplan con los objetivos del sistema.

#### **Requisitos Asociados:**

- R010: Los administradores deben poder crear nuevas tareas asignándoles nombre, descripción y puntos de recompensa.
- R011: Los administradores deben poder editar tareas existentes.
- R012: Los administradores deben poder visualizar una lista de tareas disponibles.

#### **Pasos de Ejecución:**

1. Iniciar sesión en el sistema con credenciales de administrador.
2. Acceder a la sección de gestión de tareas.

2º C.F.G.S. Desarrollo de Aplicaciones Multiplataforma. IES Padre Suárez.

Martín Mrowczynski-Van Allen Romero.

3. Crear una nueva tarea con datos específicos.
4. Editar la tarea recién creada cambiando su descripción y puntos.
5. Visualizar la lista de tareas para confirmar que los cambios se reflejan correctamente.
6. Salir de la sesión.

**Datos de Entrada:**

- Nombre de la Tarea: "Documentación del proyecto"
- Descripción Inicial: "Redactar los primeros borradores."
- Puntos: 50
- Nueva Descripción: "Actualizar los documentos del proyecto con nuevos requisitos."
- Puntos Actualizados: 75

**Condiciones Previas:**

- El administrador debe tener permisos para acceder a la gestión de tareas.
- La base de datos debe estar operativa y accesible.

**Resultado Esperado:**

- La tarea debe crearse con los datos iniciales especificados.
- La tarea debe actualizarse correctamente con la nueva descripción y los puntos ajustados.
- La lista de tareas debe mostrar los detalles actualizados de la tarea editada.

**Criterios de Aceptación:**

- El caso de prueba se considera exitoso si todas las funciones de gestión de tareas (crear, editar, visualizar) operan sin errores y reflejan los cambios correctamente en la interfaz de usuario.

**Entorno de Prueba:**

- IDE: NetBeans.
- Navegador: Chrome última versión.
- Base de Datos: MySQL en un entorno local.

**Notas Adicionales:**

- Se debe asegurar que los campos de entrada para la creación y edición de tareas validen correctamente los tipos de datos ingresados por los usuarios.

**Estado del Caso de Prueba:** **Éxito**

## Tercera fase.

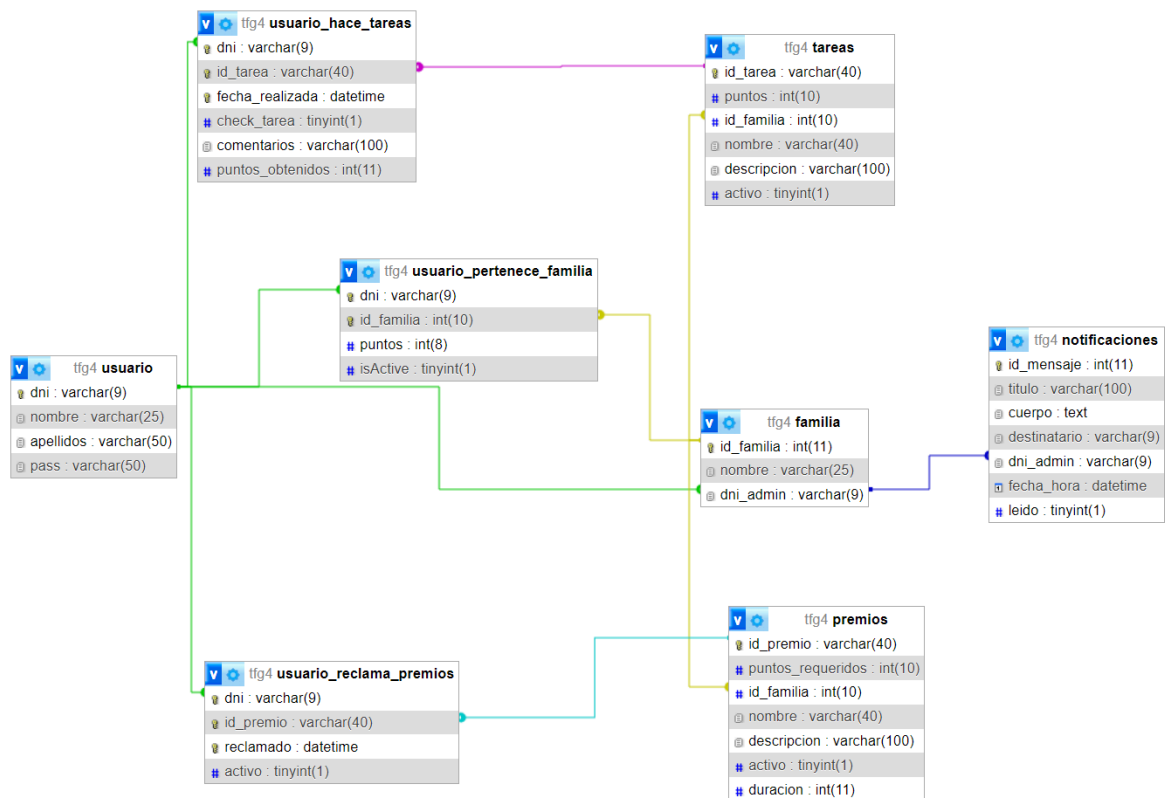
- Manual de usuario

Documento externo.

Es importante recalcar que el Manual del Usuario se ha escrito en un tono más informal con el objetivo de mantener la atención del usuario joven, un ejemplo de ello son los indicadores dibujados a mano en las imágenes o las bromas.

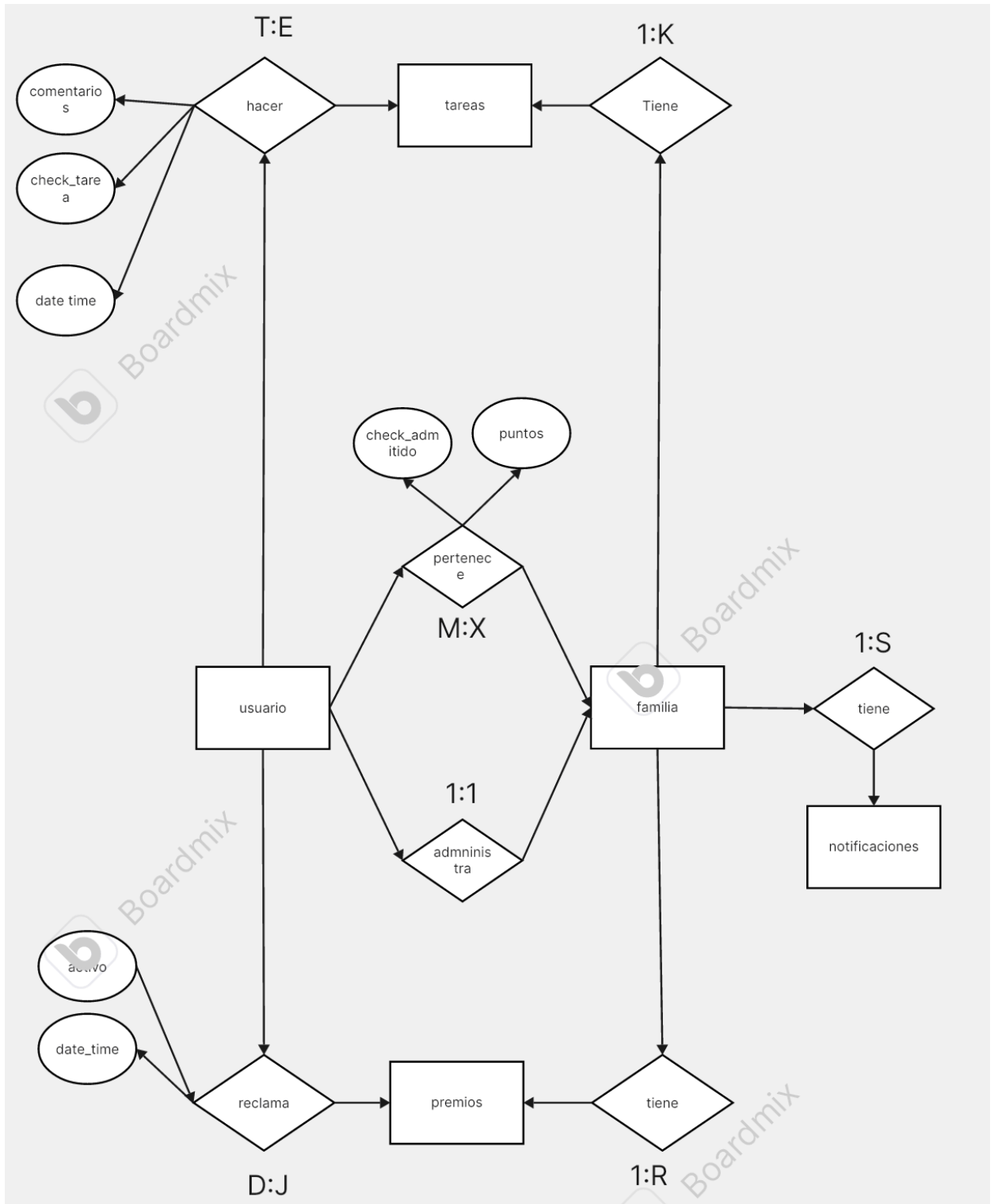
- Documento descriptivo de la base de datos

## Diseño Relacional





## Modelo E-R



## Entidades y Atributos

### 1. Usuario

- DNI
- Nombre
- Apellido
- Contraseña (pass)

### 2. Familia

- ID de Familia (id\_familia)
- Nombre
- DNI del Administrador (DNI\_admin)

### 3. Tareas

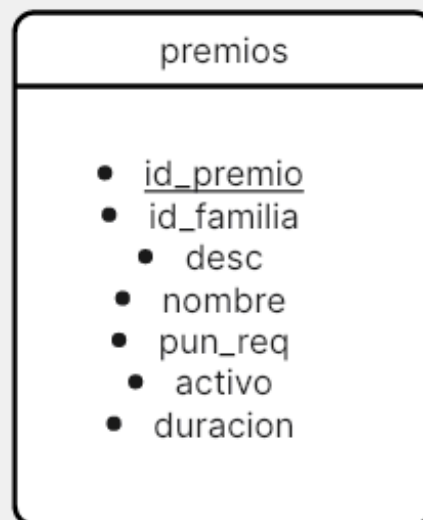
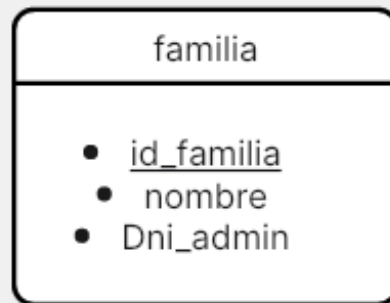
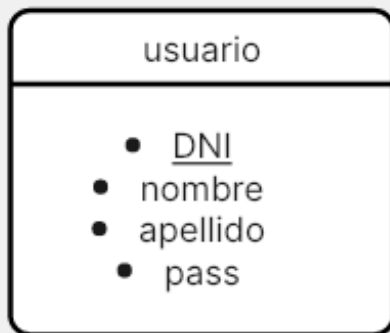
- ID de Tarea (id\_tarea)
- Puntos
- ID de Familia (id\_familia)
- Nombre
- Descripción
- Activo (indica si la tarea está activa)

### 4. Premios

- ID de Premio (id\_premio)
- ID de Familia (id\_familia)
- Nombre
- Puntos Requeridos (punt\_req)
- Activo (activo)
- Duración

### 5. Notificaciones

- ID de Mensaje (id\_mensaje)
- Título
- Descripción
- Fecha y Hora (datetime)
- Leído (indica si el mensaje ha sido leído)



## **Relaciones**

### **Familia tiene Tareas**

Una familia puede tener múltiples tareas, pero cada tarea pertenece a una sola familia

### **Familia tiene Premios**

Una familia ofrece varios premios pero cada premio solo tiene una familia.

### **Usuario pertenece a Familia**

Un usuario puede pertenecer a varias familias y una familia puede incluir a muchos usuarios.

### **Usuario administra Familia**

Un usuario administra exactamente una familia.

### **Familia tiene Notificaciones**

Una familia recibe varias notificaciones y las notificaciones pertenecen a una sola familia.

### **Usuario reclama Premios**

Un usuario puede reclamar múltiples premios y un mismo premio puede ser reclamado por varios usuarios.

### **Usuario realiza Tareas**

Un usuario puede realizar múltiples tareas y una misma tarea puede ser realizada por varios usuarios.

## **Atributos de Relación**

**Relación "usuario\_hace\_tareas" (entre usuario y tareas)**

- Comentarios (comentarios): Detalles adicionales proporcionados por el usuario sobre la realización de la tarea.
- Estado de Verificación de la Tarea (check\_tarea): Indica si la tarea ha sido revisada y verificada por un administrador.
- Fecha y Hora (fecha\_realizada): La fecha y hora exactas cuando se completó la tarea.
- Puntos Obtenidos (puntos\_obtenidos): La cantidad de puntos que el usuario ganó.

#### **Relación "usuario\_pertenece\_familia" (entre usuario y familia)**

- Puntos (puntos): Los puntos acumulados por el usuario dentro de esa familia específica.
- Estado Activo (isActive): Un indicador booleano que muestra si el usuario está activo dentro de la familia.

#### **Relación "usuario\_reclama\_premios" (entre usuario y premios)**

- Fecha Reclamado (reclamado): La fecha y hora exactas cuando el premio fue reclamado por el usuario.
- Estado Activo del Reclamo (activo): Un indicador booleano que muestra si el reclamo del premio sigue siendo válido.

## Normalización

### Primera Forma Normal (1FN)

#### **Requisitos:**

- Los valores en cada columna de una tabla deben ser atómicos (sin múltiples valores en una columna).
- Todos los valores en una columna deben ser del mismo tipo de datos.
- Cada columna debe tener un nombre único.
- El orden en que se almacenan los datos no debe importar.

La bbdd ya cuenta con la 1FN ya que todos los campos son atómicos y cada columna contiene valores del mismo tipo.

### Segunda Forma Normal (2FN)

#### **Requisitos:**

- La tabla debe estar en 1FN.
- 2º C.F.G.S. Desarrollo de Aplicaciones Multiplataforma. IES Padre Suárez.  
Martín Mrowczynski-Van Allen Romero.

- Todos los atributos no clave deben depender únicamente de la clave primaria.

La bbdd ya cuenta con la 2FN ya que los atributos no clave solo dependen de la clave primaria y se encuentra en 1FN.

## Tercera Forma Normal (3FN)

### Requisitos:

- La tabla debe estar en 2FN.
- No debe haber dependencias transitivas, es decir, los atributos no clave solo deben depender de las claves primarias y no de otros atributos no clave.

La bbdd ya cuenta con la 3FN ya que cumple todos los requisitos.

- Guía de instalación

Debido a fallos en el sistema no se ha podido ejecutar el instalador correctamente, no obstante, estará adjunto al proyecto.