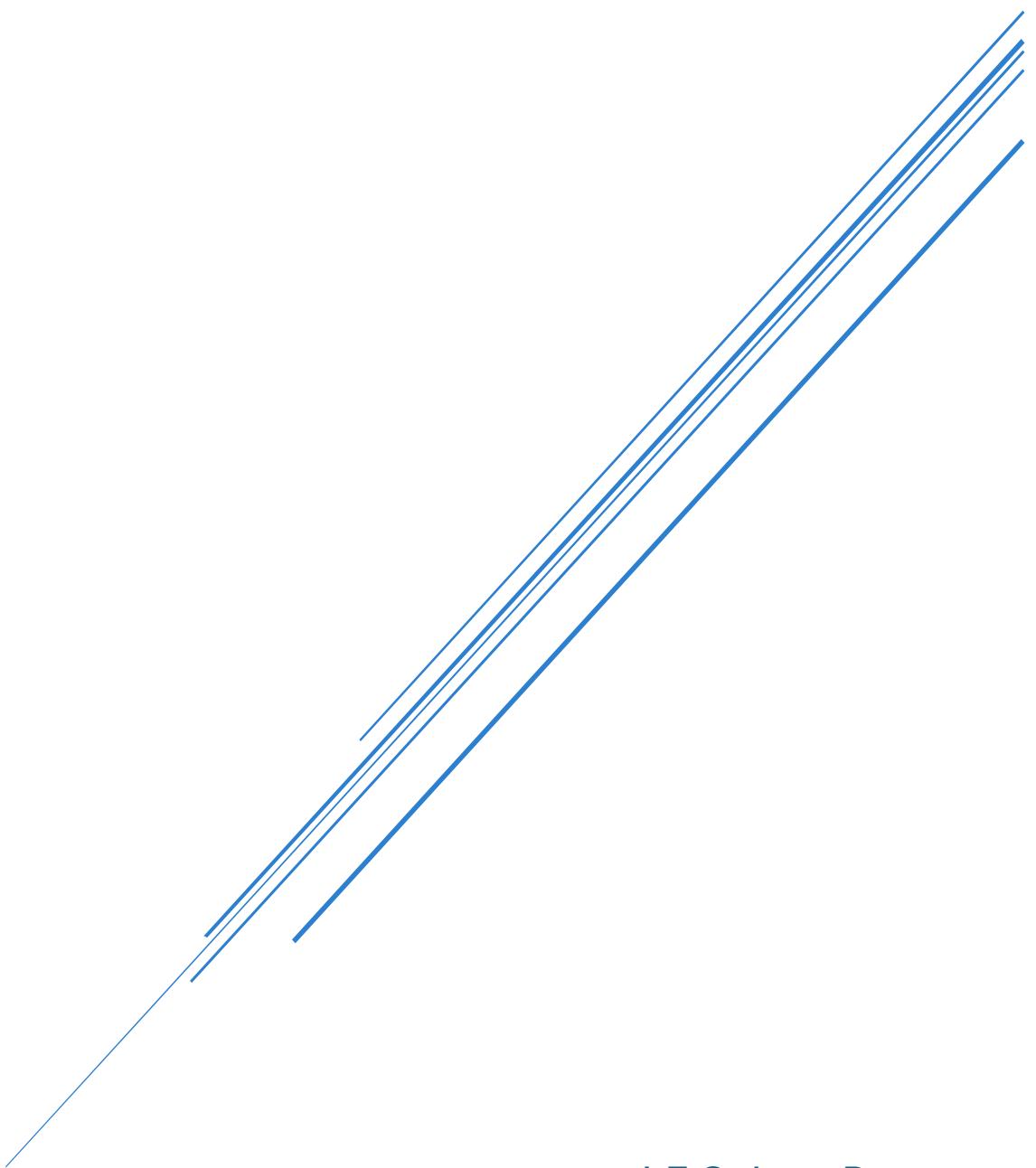


TECHNICAL DESIGN DOCUMENT

API.REST Sistema de Logros (WoW Edition)



I.E.S. Juan Bosco
Acceso a Datos

Contenido

1.	Introducción	2
2.	Diagrama de la Base de Datos (E-R)	2
3.	Especificaciones de Endpoints.....	3
4.	Modelos de Datos(Estructura JSON)	7
5.	Reglas de negocio.....	8
5.1	Personaje y Hermandad.....	8
5.2	Reglas de Progreso	8
5.3	Reglas de Borrado	8
5.4	Reglas de clase	9
5.5	Reglas de facción	9
5.6	Reglas de raza	9
5.7	Reglas de logro.....	9

1. Introducción

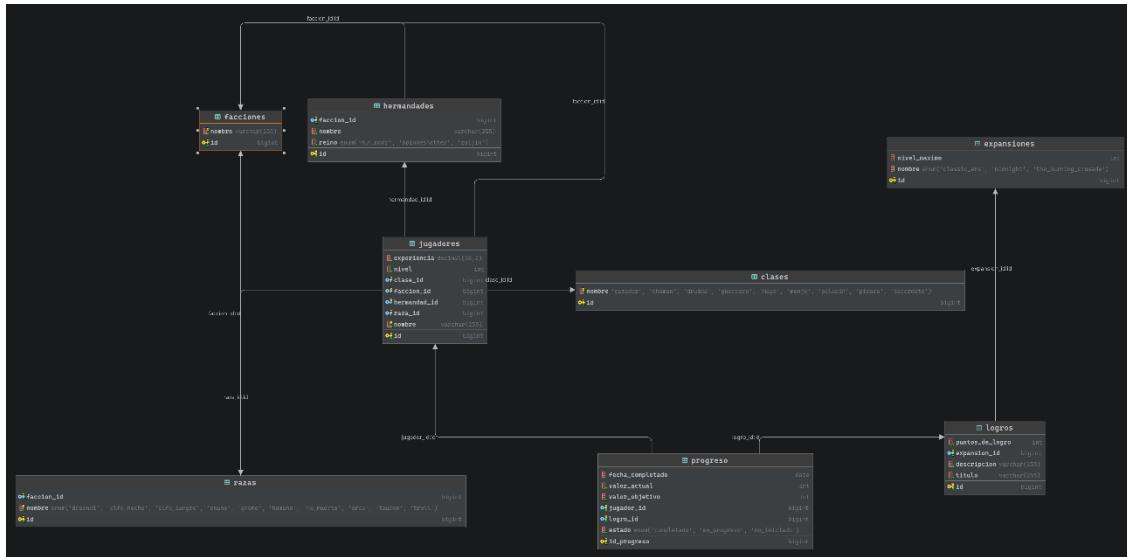
API diseñada para gestionar personajes, hermandades y el progreso de logros vinculados a las diferentes expansiones de World of Warcraft.

La API será capaz de gestionar el progreso de logros de los personajes, además de recompensar a estos personajes con logros con puntos de logros, de cual luego podremos obtener un ranking de personajes por puntos de logros.

2. Diagrama de la Base de Datos (E-R)

- **Hermandad:** Entidad maestra para agrupar personajes. (En posibles actualizaciones podrá haber logros de hermandad).
- **Jugador:** Entidad principal que representa a los jugadores que son los que obtienen puntos de logro.
- **Expansión:** Distintas versiones del juego que tienen logros específicos para cada una.
- **Logro:** Entidad que representa los logros/retos que el jugador puede conseguir.
- **Progreso:** Entidad asociativa (N:M) que registra la evolución de un jugador en un logro.
- **Clase:** Entidad que determina que clases están disponibles para los personajes
- **Raza:** Entidad que determina que razas están disponibles para los personajes y facción

- **Facción:** Entidad que determina las facciones disponibles para los jugadores y razas.



3. Especificaciones de Endpoints

Diseño de las rutas siguiendo las convenciones REST

Gestión de Jugadores

Método	Endpoint	Acción	Nota
GET	/api/jugadores	Listar personajes	Permite filtros por nombre, clase, raza y hermandad. Permite Página.
GET	/api/jugadores/{id}	Perfil jugador	
PUT	/jugadores/{id}	Editar jugador	Permite editar nombre
PUT	/jugadores/{id}/hermandad	Unirse a una hermandad	Actualiza id_hermandad en la tabla jugador
POST	/jugadores	Crear	Inicializa datos base.
DELETE	/api/jugadores/{id}	Borrar	Borra personaje y se borra de

Método	Endpoint	Acción	Nota
		personaje	la lista de personajes de la hermandad

⌚ Gestión de Hermandades

Método	Endpoint	Acción	Notas
GET	/api/hermandades	Listar todas las hermandades	Permite filtros por nombre, reino y facción Permite Paginación.
GET	/api/hermandades/{id}	Información de una hermandad	
POST	/api/hermandades	Crear una hermandad	Inicializa datos base
PUT	/api/hermandades{id}	Permite editar una hermandad	Edita el nombre de una hermandad
DELETE	/api/hermandades/{id}	Borra una hermandad	Elimina el id asociado de hermandad en la tabla jugador

🏆 Gestión de Logros

Método	Endpoint	Acción	Notas
GET	/api/logro	Listar logros	Permite filtros por expansión, titulo,

Método	Endpoint	Acción	Notas
GET	/api/logro/{id}	Listar un logro	
POST	/api/logro	Crear un logro nuevo	Inicializa datos del logro
PUT	/api/logro/{id}	Edita un logro	Edita las características de un logro
DELETE	/api/logro/{id}	Borrar un logro	

Gestión de Expansión

Método	Endpoint	Acción	Notas
GET	/api/expansiones	Listar expansiones	
GET	/api/expansiones/{id}	Obtiene una expansión por su id	
PUT	/api/expansiones/{id}	Editar expansión	Permite editar el nombre de una expansión
POST	/api/expansiones	Crear una expansión	Inicializa datos de una expansión
DELETE	/api/expansiones/{id}	Borra una expansion	

Gestión de Clases

Método	Endpoint	Acción	Notas

Método	Endpoint	Acción	Notas
GET	/api/clases	Listar clases	
GET	/api/clases/{id}	Obtiene datos de una clase en concreto	
PUT	/api/clases/{id}	Editar clase	Permite editar clases
POST	/api/clases	Permite crear una nueva clase	Inicializa datos de una clase
DELETE	/api/clases/{id}	Borra una clase	

Gestión de Razas

Método	Endpoint	Acción	Notas
GET	/api/razas	Listar razas	
GET	/api/razas/{id}	Obtiene una clase por su id	
PUT	/api/razas/{id}	Permite editar una razas	
POST	/api/razas	Crear una razas	
DELETE	/api/razas/{id}	Borra una expansion	

Gestión de Facciones

Método	Endpoint	Acción	Notas
GET	/api/facciones	Listar facciones	
GET	/api/facciones/{id}	Obtiene una faccion por su id	
PUT	/api/faccion/{id}	Permite editar una facción	
POST	/api/facción	Crear una facción	
DELETE	/api/faccion/{id}	Borra una faccion	

4. Modelos de Datos(Estructura JSON)

Recurso para jugador

```
{
  "id": 1,
  "raza": "Humano",
  "nombre": "Arthas",
  "clase": "Caballero de la Muerte",
  "nivel": 80,
  "nombre_hermandad": "ROMPO CADA TÓTEM"
}
```

Recurso para listar jugadores

```
[
  {
    "id": 1,
    "raza": "Humano",
    "nombre": "Arthas",
    "clase": "Caballero de la Muerte",
    "nivel": 80,
    "nombre_hermandad": "ROMPO CADA TÓTEM"
  },
  {
    "id": 2,
    "raza": "Orco",
    "nombre": "Thrall",
    "clase": "Mago de la Muerte",
    "nivel": 75,
    "nombre_hermandad": "EL REY DE LOS ORCOS"
  }
]
```

```

        "clase": "Chamán",
        "nivel": 80,
        "nombre_hermanadad": "WARCHIEFS"
    }
]

```

5. Reglas de negocio

5.1 Personaje y Hermandad

Un personaje solo puede pertenecer a una hermandad, para cambiarse de hermandad antes tiene que darse de baja a la actual.

Un personaje puede no pertenecer a ninguna hermandad.

Una hermandad solo puede tener miembros de una facción (Alianza o Horda).

El campo nivel no puede superar el nivel_maximo definido por la entidad Expansión. (Opcional)

El campo nivel de personaje solo puede ser un “INT” > 0.

5.2 Reglas de Progreso

El valor_actual no puede superar al valor_objetivo. Al recibir un valor superior, se debe “truncar” al valor_objetivo.

Cuando valor_actual > 0 el estado debe cambiar de “NO_INICIADO” a “EN_PROGRESO”

Cuando valor_actual == valor_objetivo el estado debe cambiar de “EN_PROGRESO” a “COMPLETADO”.

Cuando valor_actual == valor_objetivo se debe generar la fecha para el campo fecha_completado

5.3 Reglas de Borrado

Borrado en Cascada: Si se elimina un Jugador, se deben eliminar automáticamente todos registros de progresos.

Protección de Catalogo: No se puede eliminar una expansión si tiene Logros asociados.

Protección de clases: No se puede borrar una clase dada de alta+

Protección de razas: No se puede borrar una raza dada de alta

Protección de facciones: No se puede borrar una facción dada de alta

5.4 Reglas de clase

Una clase dada de alta no podrá ser borrada, se determinan las clases para siempre.

La creación de clases está limitada para ciertas razas, por ejemplo: Druidas solo pueden crearse cuando son Tauren o Elfo de la Noche.

5.5 Reglas de facción

Una facción dada de alta no podrá ser borrada, ni editada, se determinan dos únicas facciones para siempre (ALIANZA Y HORDA)

5.6 Reglas de raza

Hay razas determinadas para cada facción, no podrán borrarse facciones ya creadas.

5.7 Reglas de logro

Existen logros que serán de una expansión o de otra, el progreso de cada logro será único para cada jugador.