

INSTITUTO PROFESIONAL DUOC UC

Ingeniería informática

Proyecto Capstone

Nombre del proyecto: Dragon del bastión del norte

Integrantes: Carlos M. Benjamín H. Zuar S.

Profesor: Carlos correa

Curso: Capstone_803D

Fecha: 24 de noviembre del 2025

Índice

1. Abstract	c	3
2. Metodología implementada	c	4
3. Evidencias del proyecto		5
a. Implementación de la arquitectura	b	
b. Funciones del proyecto	z	
c. Funciones y CRUD funcionando	c	
d. Control de roles	c	
e. Diagrama de base de datos	b	
f. Evidencias visuales del proyecto		all
4. Relación proyecto con intereses profesionales		
5. Competencia de perfil de egreso		
6. Conclusiones y reflexiones		
7. Referencias		
8. Anexos		

Abstract

El presente documento detalla el proceso de diseño, desarrollo e implementación de una solución de software orientada a resolver la ineficiencia operativa de la tienda "Piedra Bruja", ubicada en Providencia, Santiago. La problemática abordada se centraba en la gestión manual de partidas de rol (*Dungeons & Dragons*), un proceso que generaba una sobrecarga administrativa para el personal y afectaba negativamente la fidelización de los clientes.

El objetivo general del proyecto fue la construcción de una aplicación web centralizada para automatizar la gestión de recursos y el seguimiento de las sesiones de juego. Para ello, se implementó una arquitectura de software escalable y desacoplada, utilizando **Next.js** para el desarrollo de una interfaz de usuario reactiva y **Django** para la lógica de negocio en el *backend*, integrando una base de datos relacional robusta mediante el uso de postgresql. El ciclo de vida del desarrollo se gestionó bajo la metodología ágil **Scrum**, lo que permitió una entrega incremental de valor al cliente, facilitando la adaptación ante cambios de infraestructura y requerimientos emergentes del cliente.

Este informe recopila evidencias técnicas y metodológicas que respaldan el cumplimiento de los objetivos, incluyendo la planificación de *Sprints*, el diseño de modelos de datos, la codificación de módulos funcionales y el despliegue productivo en servicios en la nube (**Render** y **Vercel**). Los resultados obtenidos demuestran no solo la optimización de los tiempos operativos de la organización, sino también la validación de las competencias profesionales del equipo en áreas de gestión de proyectos, arquitectura de sistemas y aseguramiento de calidad de software.

Metodología implementada

Para garantizar que el proyecto cumpliera con los requisitos operativos de "Piedra Bruja", el equipo adoptó la metodología ágil Scrum. Esta decisión permitió gestionar el desarrollo de manera iterativa, entregando valor real al cliente paso a paso. Además, este enfoque brindó la flexibilidad necesaria para trabajar con una arquitectura técnica dividida (Frontend en Next.js y Backend en Django) y adaptarse a los cambios de infraestructura que surgieron durante el proceso.

El trabajo se organizó en ciclos cortos denominados *Sprints*, cuyo objetivo siempre fue finalizar con un incremento de software funcional. Al inicio de cada ciclo, se seleccionaron y priorizaron las historias de usuario según su importancia para el negocio. Para asegurar la calidad técnica, el equipo estimó la complejidad de cada tarea y aplicó buenas prácticas de control de versiones en GitHub (mediante *Pull Requests*), protegiendo así la integridad del código. Al finalizar la semana, se presentaron los avances al cliente para recibir retroalimentación inmediata.

Para la gestión y trazabilidad de las tareas, se utilizó la herramienta Trello. A través de tableros de control visuales (divididos en estados como "Por hacer", "En proceso" y "Terminado"), el equipo

mantuvo un seguimiento claro del *Sprint Backlog*, asegurando que todos los integrantes conocieran el estado del desarrollo en tiempo real.

La utilidad de esta metodología se hizo evidente al enfrentar problemas críticos de infraestructura. Cuando se detectaron limitaciones severas en las bases de datos gratuitas, la capacidad de reacción que permite Scrum facilitó un cambio de estrategia rápido. El equipo pudo migrar hacia una solución de pago y ajustar la configuración de los servicios de despliegue (Render y Vercel) sin detener el avance general del proyecto. En conclusión, esta implementación rigurosa no solo organizó el trabajo colaborativo, sino que garantizó la entrega de una solución robusta y validada continuamente por el cliente.

Evidencias del proyecto

A) Implementación de la arquitectura

La arquitectura del sistema "Piedra Bruja" fue implementada siguiendo estrictamente el diseño de Tres Capas presentado en las etapas de planificación, consolidando una separación clara y funcional entre la capa de presentación, la lógica de negocio y la gestión de datos.

El backend fue desarrollado utilizando Django REST Framework, aprovechando su estructura basada en ViewSets y Serializers para organizar el sistema en módulos tales como la gestión de personajes, el inventario de personajes, tiendas de objetos, etc. Esta organización permitió mantener un código ordenado y seguro, implementando lógica compleja como transacciones para asegurar la integridad de los datos en operaciones críticas, por ejemplo, la donación de objetos al inventario compartido o el cálculo de progresión de rangos.

Durante el desarrollo, la persistencia de datos se implementó utilizando PostgreSQL. Se utilizaron herramientas como DBeaver para la gestión y población de datos maestros en entornos locales, facilitando la validación del modelo relacional. Para el despliegue final, el sistema migró exitosamente a una infraestructura en la nube alojada en Render, manteniendo el mismo esquema relacional definido en los modelos de Django. Esta transición garantizó la disponibilidad y estabilidad necesarias para el uso real de la plataforma.

Las rutas y endpoints de la API fueron consumidos por una interfaz moderna desarrollada en Next.js. Se implementó una arquitectura de componentes reutilizables que interactúan con el backend mediante peticiones protegidas por autenticación JWT. Este enfoque permitió ofrecer una experiencia de usuario dinámica y diferenciada según el rol, logrando una navegación fluida y reactiva sin necesidad de recargar la página completa.

La correcta orquestación entre el frontend en Next.js, la API en Django y la base de datos PostgreSQL en la nube confirma la implementación exitosa de la arquitectura propuesta. El sistema resultante es modular, escalable y cumple con los objetivos técnicos de digitalizar y centralizar la gestión de partidas de Dungeons & Dragons de manera eficiente.

B) Funciones del proyecto

En esta sección se nombraran las distintas funcionalidades que se implementaron en el sistema:

- Permitir el registro y autenticación de usuarios mediante un sistema de login seguro y gestión de sesiones activas
- Establecer la separación de roles entre usuarios estándar y administradores, definiendo accesos diferenciados para cada perfil.
- Proveer al administrador un panel de gestión con navegación centralizada (navbar) que facilite el acceso a los distintos módulos administrativos.
- Permitir al administrador crear, editar y eliminar objetos, recetas, tiendas, especies, subclases, dotes y demás recursos propios del sistema.
- Facilitar al administrador la gestión de personajes existentes, incluyendo visualización, actualización de datos y control del inventario.
- Habilitar al administrador para asignar y retirar ítems de personajes según las necesidades del sistema.
- Proporcionar herramientas de administración para supervisar actividades generales y mantener la integridad de los datos.
- Permitir a los usuarios crear y personalizar su personaje, definiendo atributos y opciones iniciales.
- Habilitar a los usuarios para comprar ítems en las tiendas disponibles dentro del sistema.
- Permitir que los usuarios investiguen objetos con el fin de desbloquear recetas asociadas.
- Implementar un sistema de crafteo basado en tiradas de dados, donde el usuario debe alcanzar un número determinado de éxitos según la dificultad (DC) del objeto, considerando bonificaciones obtenidas por habilidades o características del personaje.

- Permitir la realización de actividades adicionales, como "irse de juerga", orientadas a obtener mejoras menores o aumentar el respeto ante tiendas o NPC.
- Mantener registros de acciones relevantes realizadas por los usuarios y administradores para asegurar trazabilidad y consistencia del sistema.
- Proporcionar una interfaz web clara e intuitiva que permite navegar de forma eficiente entre las funciones disponibles tanto para administradores como para usuarios estándar.

C) Funciones y CRUD funcionando

El funcionamiento del código en el sistema sigue un proceso estructurado que inicia en el *backend* con la definición de **Modelos** en Django, los cuales permiten la creación de tablas relacionadas en la base de datos PostgreSQL. Para habilitar la comunicación entre aplicaciones, se utilizan **Serializadores** que traducen estas entidades complejas a formato JSON, mientras que las **Vistas** (*Views*) encapsulan la lógica de negocio exponiéndose a través de **URLs** (*endpoints*) seguras. En el *frontend*, la aplicación Next.js consume estos servicios mediante peticiones HTTP asíncronas, integrando los datos en la interfaz a través de tablas dinámicas para la visualización y formularios controlados que capturan y validan la información del usuario.

Este esquema técnico sustenta el núcleo principal de la plataforma, basado en la implementación de sistemas CRUD (*Create, Read, Update, Delete*) robustos diseñados para administrar las entidades críticas del juego. Siguiendo el *Sprint Planning*, la implementación abarcó tres áreas clave:

- **Gestión de Economía y Recursos:** Se desarrollaron controladores lógicos para la administración de "Objetos" y "Tiendas". Esto permite no solo crear ítems, sino realizar transacciones complejas de compra de objetos, asegurando mediante validaciones estrictas en el *Backend* que los personajes posean el oro y los requisitos necesarios antes de persistir las transacciones en la base de datos.
- **Sistemas de Profundización RPG:** Para soportar la escalabilidad de las reglas de *Dungeons & Dragons*, se implementaron mantenedores para entidades estáticas como "Especies", "Clases" y "Dotes". Estos módulos alimentan dinámicamente los componentes del *Frontend*, garantizando que la creación de personajes se adapte siempre a las reglas definidas por los administradores.

Finalmente, cada operación CRUD funciona mediante la exposición de la API REST bajo un flujo de trabajo que asegura la **integridad referencial**: por ejemplo, si se intenta eliminar una "Receta de Crafteo", el sistema verifica automáticamente que no existan procesos activos dependientes de ella, previniendo inconsistencias y corrupción de datos en el sistema.

D) Control de roles

Para garantizar la seguridad de la información y la correcta repartición de funciones dentro de la plataforma, se implementó un sistema de control de acceso basado en roles. Los cuales se dividen en dos perfiles claramente diferenciados, asegurando que la gestión administrativa no interfiera con la experiencia de usuario final.

- **Rol de Administrador:** Este perfil posee privilegios elevados orientados a la gestión económica y normativa del juego. Los administradores tienen la facultad de **modificar valores críticos** en las fichas de los personajes, tales como el Oro, Tiempo Libre, Puntos de Tesoro y *Checkpoints*, controlando así directamente la progresión del nivel de los jugadores. Además, son los responsables de alimentar los datos maestros del sistema; a través de CRUDs específicos, definen las reglas, objetos disponibles, tiendas y opciones (Clases/Especies) que estructuran el entorno en el que interactúan los usuarios.
- **Rol de Jugador:** Diseñado para la operatividad diaria, este perfil permite a los usuarios **gestionar sus activos** dentro de los límites definidos por los clientes. Los jugadores cuentan con herramientas para la creación de personajes, gestión de inventario e intercambio de objetos. Asimismo, tienen acceso a mecánicas para incrementar sus recursos básicos (como trabajar para obtener oro o tiempo libre), fomentando una participación activa en la economía del sistema sin comprometer la integridad de las reglas globales.

E) Diagrama de base de datos

Diagrama relacional del sistema creado mediante la opción de diagramar del programa Dbeaver



Visualización de las tablas en el programa Dbeaver

Tablas	
>  api_bonusproficiencia	40K
>  api_classfeature	48K
>  api_classresource	48K
>  api_competenciaherramienta	56K
>  api_dndclass	96K
>  api_dndclass_skill_choices	32K
>  api_dndfeat	128K
>  api_dndsubclass	80K
>  api_dndsubclass_skill_choices	72K
>  api_featfeature	64K
>  api_habilidad	56K
>  api_historialtirada	40K
>  api_historialtiradainvestigacion	40K
>  api_ingredientes	56K
>  api_inventario	72K
>  api_inventarioparty	88K
>  api_npc	112K
>  api_objeto	1,5M
>  api_objetotienda	72K
>  api_pagorango	56K
>  api_party	80K
>  api_party_miembros	72K
>  api_personaje	88K
>  api_personaje_dotes	72K
>  api_personajeeleccion	72K
>  api_proficiencia	56K
>  api_progresoinvestigacion	104K
>  api_progresoreceta	72K
>  api_progresotrabajo	72K
>  api_receta	56K
>  api_recetadesbloqueada	72K
>  api_relacionnpc	72K
>  api_species	96K
>  api_subclassfeature	64K
>  api_subclassresource	48K
>  api_tienda	64K
>  api_tptransaction	56K
>  api_trabajo	80K
>  api_trabajorealizado	56K
>  api_trait	64K
>  auth_group	24K
>  auth_group_permissions	32K
>  auth_permission	88K
> auth_user	64K
> auth_user_groups	32K
> auth_user_user_permissions	32K
> django_admin_log	32K
> django_content_type	40K
> django_migrations	32K
> django_session	32K

F) Evidencias visuales del proyecto

A continuación se muestran las evidencias funcionales con una demostración del uso de cada funcionalidad principal de la página

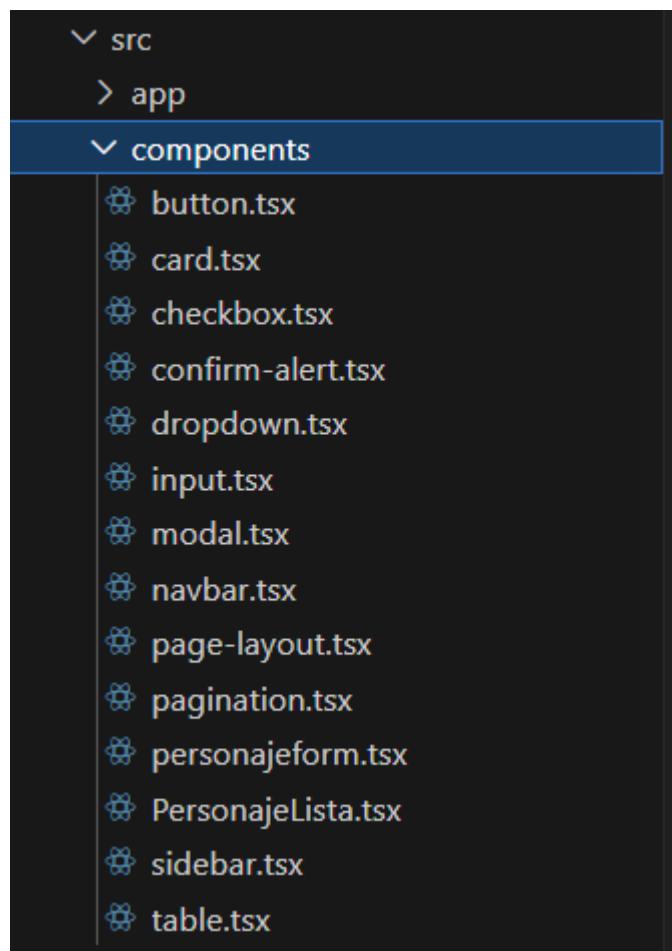
Mediante el siguiente documento se puede apreciar el código:

 Evidencias final

Registro en entorno de desarrollo

<https://www.youtube.com/watch?v=RRdhbK3LnAE>

Componentes de react



Página de activación de cuentas



Integración de Crud de objetos

<https://www.youtube.com/watch?v=N0qM8Jt5ZkE>

Integración de Crud de tiendas

<https://www.youtube.com/watch?v=v19Dcs1Om0A>

Integración de compras

<https://www.youtube.com/watch?v=etJQMz6QEIE>

Integración de Crud trabajos

<https://youtu.be/Ynbpd-30GQw>

Integración de Crud trabajos (Modificado)

<https://youtu.be/TaNw3DZMDYI>

Integración de Crud creacion personaje

<https://www.youtube.com/watch?v=B6JDTNLrTvs>

Integración CRUD de clases

<https://youtu.be/257BkWHqzrl>

Integración CRUD de especies

<https://www.youtube.com/watch?v=LnY1dY12xvw>

Integración CRUD de subclases

<https://www.youtube.com/watch?v=OmCoumTLEuY>

Creación de personajes detallada

<https://youtu.be/7abdV8UWcel>

Irse de juerga

<https://youtu.be/UmfIxdNNDKU>

Tienda de tesoro

<https://youtu.be/AYrq7DZJDWA>

Crafeo de items usuario

<https://www.youtube.com/watch?v=stJOW1c9rd0>

Investigación de ítems

<https://www.youtube.com/watch?v=35yTc7AadBE>

CRUD Dotes

<https://youtu.be/CkeNRliGb2s>

Funcion de Grupos

<https://youtu.be/gGjXrMtYAD8>

Aporte del proyecto a tus intereses profesionales

Carlos Manzo

El desarrollo de este proyecto ha sido de gran ayuda para mi crecimiento profesional, permitiéndole avanzar de la programación básica para enfocarme en la **arquitectura y el despliegue de soluciones reales**. Uno de los mayores aportes fue la experiencia práctica en la implementación de una arquitectura de software desacoplada; aprendí a integrar y comunicar aplicaciones independientes (*Frontend* y *Backend*) mediante protocolos seguros.

Asimismo, la adopción de **Next.js** como nueva herramienta tecnológica basada en React expandió mi stack técnico, obligándome a comprender conceptos avanzados de renderizado. Finalmente, este proyecto redefinió mi enfoque hacia la gestión del desarrollo: aprendí a priorizar la programación no solo desde la técnica, sino alineándose estrictamente con las necesidades del cliente, transformando requerimientos de negocio en funcionalidades de código tangibles y útiles.

Benjamin Huentemilla

Poder trabajar en este proyecto me dio una nueva visión sobre lo que realmente me interesaba, pues pese a que la programación web no entraba dentro de mis propios intereses a medida que íbamos avanzando y programando, también aprendímos nuevas tecnologías, las que ami en un principio pensaba que no me generaron mucho interés, pero actualmente agradezco haberlas aprendido y aplicado más en un caso real, por lo que al final considero que este proyecto fue de gran ayuda para ampliar mis conocimientos e intereses a futuro sobre a qué dedicarme

Zuar Sepulveda

La realización de este proyecto no me parecía interesante pero a medida que el proyecto avanzaba nuevos descubrimientos y desafíos emergieron, a pesar que la documentación no es uno de mis puntos fuertes realizar estos documentos y la realización del trello me abrió la mente a cual importante es. también el aprendizaje e incorporación de nuevas tecnologías que expande el abanico de conocimientos, me permitió usar los que ya poseía y mejorarlo.

Competencias del perfil de egreso

El desarrollo del proyecto para la tienda "Piedra Bruja" permitió consolidar las competencias clave del perfil de egreso, transformando conocimientos que previamente solo conocíamos en la teoría en una solución tecnológica real.

En primer lugar, se abordó la competencia de **gestionar proyectos informáticos ofreciendo alternativas para la toma de decisiones**. Esto se materializó a través de la metodología Scrum, donde el equipo no solo planificó tareas, sino que debió evaluar y decidir estratégicamente sobre la infraestructura tecnológica. Un ejemplo claro fue la gestión del incidente con la base de datos gratuita; el equipo analizó la situación y ofreció una alternativa viable (migración a servicios de pago) para cumplir con los requerimientos de la organización, asegurando la continuidad operativa del negocio.

Asimismo, el proyecto exigió **construir modelos de datos escalables y desarrollar una solución de software sistematizada**. La complejidad de las reglas del sistema de juego *Dungeons & Dragons* requirió un diseño de base de datos relacional robusto (PostgreSQL) capaz de soportar el crecimiento futuro de la tienda. Esto se integró mediante una arquitectura desacoplada (Next.js y Django) que cumple con el estándar de la industria, facilitando el mantenimiento y la escalabilidad del sistema, asegurando el logro de los objetivos técnicos y comerciales del proyecto.

Finalmente, se cumplió con la competencia de **realizar pruebas de certificación y validación de productos**. Mediante la implementación de validaciones lógicas en el código (como la integridad en las transacciones de compraventa) y las revisiones periódicas con el cliente (*Sprint Reviews*), se aplicaron buenas prácticas para asegurar que el software entregado no solo fuera funcional, sino que respondiera fielmente a las necesidades operativas detectadas al inicio del proyecto.

Conclusiones y reflexiones finales

El desarrollo del proyecto “Dragon del bastión de norte” fue una experiencia que permitió al equipo crecer tanto técnica como profesionalmente. A medida que avanzaron, cada integrante descubrió nuevas herramientas, tecnologías y formas de trabajo que ampliaron su visión sobre el desarrollo de software más allá de la programación básica.

El trabajo con una arquitectura desacoplada, el uso de Next.js y Django, y la construcción de un modelo de datos complejo fueron aprendizajes claves que ayudaron al equipo a comprender cómo se desarrollan soluciones más reales y escalables. Además, la documentación y el uso de herramientas de gestión como Trello les demostraron la importancia de organizar y justificar cada decisión tomada durante el proceso.

La metodología Scrum facilitó la planificación y la toma de decisiones, especialmente cuando surgieron problemas como la migración de la base de datos. Estas situaciones reforzaron la capacidad del equipo para adaptarse y buscar alternativas viables.

Finalmente, las pruebas, validaciones y revisiones con el cliente permitieron asegurar que el resultado final cumpliera con lo solicitado. En conjunto, el proyecto permitió consolidar competencias del perfil de egreso y fortaleció los intereses profesionales del grupo, dejando una experiencia valiosa para futuros desafíos.