

**S.I.G.P.D.**

Full Stack

Dreams Magic Wizard

| **Rol** | **Apellido** | **Nombre** | **C.I** | **Email** |
| --- | --- | --- | --- | --- |
| Coordinador | Espinosa | Jonathan | 4.563.848-6 | jonathan.espinosa@estudiante.ceibal.edu.uy |
| Sub-Coordinador | Estévez | Leandro | 4.438.718-9 | lelo.estevez24@gmail..com |
| Integrante 1 | Perdomo | Martin | 4.986.528-7 | martin.perdomo@estudiante. ceibal.edu.uy |
| Integrante 2 | Uval | German | 5.476.329-6 | valentin.uval.barboza@gmail. com |
| Integrante 3 | Kramarchuk | Robert | 4.828.299-3 | kramarchuk108@gmail.com |

**Docente: Mendez Bautista, Diego Alejandro**



# **Índice**

[**Índice 1**](#_2wvxf5ww2gb5)

[**Justificación de Selección de Tecnologías para el Proyecto Draftosaurus 2**](#_9yggnwwipifx)

[1. Introducción 2](#_gmt1fi9pc1al)

[2. Requerimientos Generales del Proyecto 2](#_pe8081uq6s8v)

[3. Tecnologías Seleccionadas y Justificación 2](#_ygnp4dkqzich)

[3.1. HTML (HyperText Markup Language) 2](#_c9b5htqhg1fj)

[3.2. CSS (Cascading Style Sheets) 3](#_1pm8hdvdrb9e)

[3.3. JavaScript (JS) 3](#_3f576p9g69ew)

[3.4. Bootstrap (Framework Frontend) 3](#_uqegg84slhre)

[3.5. SQL (Structured Query Language) 4](#_7utpfgy3eff7)

[**Selección de herramientas para control de versiones y colaboración 5**](#_aqg9jkpv8j5)

[Herramienta elegida: Git + GitHub 5](#_x0dnvevrh1yx)

[Convenciones para Commits y Mensajes de Commit 5](#_bp1ry9zgjgrd)

[Objetivo: 5](#_vd5tsgvrubk5)

[Estructura detallada del mensaje de commit 6](#_ddikxscf7o5e)

[1. Encabezado (Header) Obligatorio 6](#_f9w10m91xe8k)

[2. Cuerpo (Body) Opcional pero recomendable 6](#_kwev7qlhk4iv)

[Reglas adicionales: 6](#_dnyg6y6a0jnm)

[Beneficios de seguir esta convención: 6](#_5wocgnkfmlby)

[Conclusión 7](#_ulh74nrd967)

# **Justificación de Selección de Tecnologías para el Proyecto Draftosaurus**

## **1. Introducción**

Este documento tiene como objetivo justificar la elección de tecnologías utilizadas para el desarrollo del proyecto **Draftosaurus**, en el marco del programa educativo de UTU. La selección de herramientas responde a los requerimientos funcionales y técnicos del sistema, optimizando el desarrollo, el diseño y la gestión de datos.

## **2. Requerimientos Generales del Proyecto**

El proyecto consiste en una aplicación web interactiva que representa el juego de mesa **Draftosaurus**, incluyendo:

* Interfaz amigable para los usuarios.
* Interactividad en la selección de piezas de dinosaurios.
* Diseño adaptable (responsive) para distintos dispositivos.
* Gestión de datos sobre jugadores y partidas.
* Almacenamiento de información persistente (base de datos).

## **3. Tecnologías Seleccionadas y Justificación**

### **3.1. HTML (HyperText Markup Language)**

**Uso:**

* Estructuración básica de la aplicación web.
* Definición de elementos de la interfaz (botones, formularios, tablas, etc.).

**Justificación:** HTML es el lenguaje estándar para la creación de contenido web. Permite establecer la estructura base de las distintas secciones del juego, como la pantalla de inicio, la selección de dinosaurios, y la visualización de partidas.

### **3.2. CSS (Cascading Style Sheets)**

**Uso:**

* Definición de estilos visuales.
* Personalización de colores, fuentes y distribución de elementos.

**Justificación:** CSS permite adaptar la estética del juego al diseño deseado, mejorando la experiencia visual del usuario. Además, facilita la adaptación responsive del sitio en combinación con Bootstrap.

### **3.3. JavaScript (JS)**

**Uso:**

* Lógica de interacción entre el usuario y la interfaz.
* Validación de datos en el navegador.
* Manipulación del DOM (Document Object Model) en tiempo real.

**Justificación:** JavaScript es fundamental para manejar la dinámica del juego Draftosaurus. Permite detectar las selecciones de los jugadores, realizar cálculos en el momento (como la puntuación), y actualizar la interfaz sin necesidad de recargar la página.

### **3.4. Bootstrap (Framework Frontend)**

**Uso:**

* Creación rápida de componentes visuales como botones, modales, alertas y navegación responsive.
* Implementación de un diseño adaptable sin necesidad de escribir CSS desde cero.

**Desventaja:**

Una de las principales desventajas de utilizar Bootstrap es que, si no se realiza una personalización adecuada de los estilos, las páginas web tienden a tener un diseño visual genérico o similar a otros sitios que también utilizan este framework. Esto puede afectar la originalidad y la identidad visual del proyecto si no se complementa con modificaciones de CSS propio o personalización de los componentes.

**Justificación:** Bootstrap es un framework liviano que facilita el diseño responsive y profesional. Su uso reduce tiempos de desarrollo y garantiza compatibilidad entre distintos navegadores y dispositivos móviles. Además, proporciona componentes predefinidos que optimizan la experiencia de usuario en el juego.

### **3.5. SQL (Structured Query Language)**

**Uso:**

* Almacenamiento de datos de los jugadores, partidas y resultados.
* Consulta y gestión de información desde la base de datos.

**Justificación:** SQL es el lenguaje estándar para gestión de bases de datos relacionales. En este proyecto, se utilizará para guardar información persistente sobre las partidas y jugadores. La elección de SQL es coherente con los contenidos del programa de UTU, además de ser una tecnología robusta y ampliamente utilizada en el desarrollo de aplicaciones.

# **Selección de herramientas para control de versiones y colaboración**

## **Herramienta elegida: Git + GitHub**

| **Herramienta** | **Uso** | **Ventajas** |
| --- | --- | --- |
| **Git** | Control de versiones local | Permite guardar el historial de cambios, volver a versiones anteriores y trabajar de forma organizada. |
| **GitHub** | Repositorio remoto y colaboración | Permite el trabajo en equipo, subir el proyecto a la nube y colaborar entre varios desarrolladores, manteniendo control de cambios y ramas. |

**Beneficios clave:**

* Facilita el trabajo colaborativo entre compañeros.
* Evita la pérdida de información.
* Permite documentar el proceso de desarrollo (commits).
* Es estándar en la industria.

## **Convenciones para Commits y Mensajes de Commit**

### **Objetivo:**

Establecer un estándar claro para los mensajes de commit, asegurando que el historial de cambios del proyecto sea **claro**, **legible** y **fácil de mantener**.

## 

## **Estructura detallada del mensaje de commit**

### **1. Encabezado (Header) Obligatorio**

* **Formato:** <tipo>(<área>): <mensaje corto>
* **Máximo:** 72 caracteres.
* **En presente, modo imperativo:** (Ejemplo: "añadir", no "añadido" o "se añadió").

### **2. Cuerpo (Body) Opcional pero recomendable**

Explicación más detallada del **qué** y **por qué** del cambio.

Explica la **motivación del cambio** y **cómo se solucionó** el problema.

Si corresponde, describe **limitaciones conocidas** o **aspectos pendientes**.

## **Reglas adicionales:**

* No usar mensajes genéricos tipo:  
   *"arreglos varios", "cambios", "update"*
* Si el cambio es grande o complejo, el **cuerpo del commit es obligatorio**.
* Cada commit debe representar **un único propósito claro**.

## **Beneficios de seguir esta convención:**

* Historial de commits **más claro y navegable**.
* Facilita el uso de herramientas como **Conventional Changelog**, **semantic versioning**, o **automatización de despliegues**.
* Mejora la **colaboración entre equipos**.

## 

## **Conclusión**

La selección de **HTML**, **CSS**, **JavaScript**, **Bootstrap** y **SQL**, junto con **Git y GitHub** para el control de versiones, responde a los requerimientos técnicos del proyecto Draftosaurus, cumpliendo además con las tecnologías incluidas en el plan de estudios de UTU. Esta combinación permite lograr una aplicación funcional, accesible y adaptable a diferentes dispositivos, garantizando un flujo de trabajo profesional y organizado.