



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2026 — VISUALIZACIÓN DE INFORMACIÓN

Hito 2

Tema: Codificaciones visuales y utilidades de D3.js

Tiempo estimado: 10 horas

Fecha de entrega: **22 de octubre** a las **20:00** hrs

Contexto

FIFA es una saga de videojuegos de fútbol que se publican anualmente y son desarrollados por **EA Sports**. Se destaca por contener la licencia oficial de la **FIFA** y poseer diversas licencias exclusivas. Esto le permite incluir equipos de ligas de todo el mundo, además de estadios y jugadores reales como personajes dentro del juego.

Uno de sus modos de juego más popular es el **Fifa Ultimate Team (FUT)**, en el cuál los jugadores son representados por tarjetas que reflejan un resumen de sus estadísticas y cualidades. Tu tarea será presentar los datos de los jugadores disponibles en una edición específica del juego mediante una visualización implementada con **D3.js**.

Dataset

El *dataset* a usar es `fifa_20_data.csv`¹, el cual puedes encontrar en el siguiente enlace de **Canvas**. Este archivo de extensión CSV contiene la información sobre los **jugadores** incluidos en la edición 20 del juego. En específico, las columnas que lo componen son:

Columna	Explicación
NAME	Nombre del jugador.
CLUB	Equipo al cual pertenece el jugador.
LEAGUE	Liga en la cual compite el jugador.
POSITION	Posición en la cual juega el jugador.
TIER	Nivel del jugador (Oro, Plata o Bronce).
RATING	Promedio ponderado de las estadísticas de un jugador. Varía dentro del rango [1, 99]
PACE	Promedio de atributos relacionados a velocidad. Varía dentro del rango [1, 99]
SHOOTING	Promedio de atributos relacionados a tiros al arco. Varía dentro del rango [1, 99]
PASSING	Promedio de atributos relacionados a pases. Varía dentro del rango [1, 99]
DRIBBLING	Promedio de atributos relacionados a regate. Varía dentro del rango [1, 99]
DEFENDING	Promedio de atributos relacionados a defensa. Varía dentro del rango [1, 99]
PHYSICAL	Promedio de atributos relacionados al físico. Varía dentro del rango [1, 99]

Nota: El promedio **RATING** no corresponde sólo al de las seis estadísticas presentadas. Es el promedio

¹Si bien esta es la **fuentes** original, se le ha hecho una pequeña limpieza para eliminar íconos y jugadores duplicados. el preprocesamiento realizado lo pueden encontrar en **este archivo**.

ponderado de [30 atributos](#), según la posición en la que juegan (ej. un defensa tiene mayor ponderación en *tackles* que un delantero).

1. Visualización de jugadores de FIFA

En este hito, deberás crear una visualización que permita presentar los datos de los jugadores presentes en el juego y cumpla con todos los requisitos que se señalarán en esta sección.

1.1. Tarjetas de jugadores [3 pts]

Para comenzar, tendrás que crear una tarjeta que represente a un único jugador (ítem) del *dataset*, el objetivo debería seguir la idea del siguiente ejemplo:



Figura 1: Ejemplo de tarjeta de jugador

Para lograr esto, tendrás que implementar los siguientes componentes:

1.1.1. Spider Chart [2 pts]

Para representar los seis atributos relacionados a las competencias físicas de un jugador (PAC, SHO, PAS, DRI, DEF, PHY), se te solicita hacerlo a través de un [Spider o Radar Chart](#).

Un *spider chart* consiste en un diagrama que posee una cantidad variable de ejes, los cuales se distribuyen como radios a lo largo de una circunferencia. Sobre estos ejes se ubica un polígono que tiene un vértice sobre cada uno de los ejes creados, siendo su posición sobre dicho eje el canal que codifica un dato correspondiente al eje.

En este caso en específico, cada uno de los seis atributos mencionados deberá ser representado por un **eje** del gráfico. Donde la posición de un vértice en dicho eje será dictada por el valor que tiene el jugador para el atributo correspondiente.

Por otro lado, el rango de valores mostrado en los ejes debe ser el mismo a lo largo de todas las tarjetas. Si no, comparar polígonos y posiciones en ejes entre tarjetas no será efectivo, al usar distintos rangos de valores.

(a) Nombre de cada estadística

En cada extremo del polígono deberá ponerse una abreviación del atributo que representa ese vértice, con el fin de poder identificar a qué atributo pertenece cada eje.

(b) Color de polígono

El color del polígono dependerá del atributo POSITION. En específico, este atributo puede ser agrupado en tres categorías, y cada una de esas categorías identificadas con un color:

Posiciones	Categoría	Color
CB, RB, LB, LWB, RWB	Defensa	Azul
CM, CAM, CDM, LM, RM	Mediocampista	Verde
LF, CF, RF, ST, RW, LW	Delantero	Rojo

Los colores no deben ser exactamente las tonalidades presentadas en la tabla, puedes escoger variaciones de los matices presentados.

1.1.2. Tarjeta [1 pt]

Una vez implementado el *spider chart* anteriormente descrito, este deberá ser contenido en una tarjeta **rectangular**, la cuál además deberá indicar mediante texto el RATING, nombre, equipo y liga de cada jugador.

La forma en que ubiques estos tres elementos quedará a tu criterio, siempre y cuando sea consistente en todas las tarjetas y sea legible. Por ejemplo, en la Figura 1 se muestra un ejemplo para el jugador de nombre Lionel Messi. Su RATING es de 95, su equipo es FC Barcelona y su liga es Spain Primera Division.

Otras características que debe tener la tarjeta son:

(a) Color de fondo

El color de la tarjeta dependerá del atributo RATING, y se regirá según el siguiente *colormap* segmentado:

RATING	Color
$75 \leq \text{RATING} \leq 99$	Oro
$65 \leq \text{RATING} < 75$	Plata
$01 \leq \text{RATING} < 65$	Bronce

Los colores no deben ser exactamente las tonalidades presentadas en la tabla, puedes escoger variaciones de los matices presentados.

1.2. Layout de tarjetas en página [3 pts]

Una vez realices la implementación de la tarjeta individual, deberás crear una página que permita visualizar las tarjetas de todos los jugadores de forma **clara** y **ordenada**. Para esta parte podrás hacer uso de HTML, CSS y JavaScript.

1.2.1. Separación de secciones [0.5 puntos]

La página debe presentar una separación clara en dos secciones, a la izquierda deberán mostrar una **barra de navegación fija**, mientras que a la derecha el cuadro principal con las tarjetas. El detalle de que

debe tener cada sección se explica a continuación:

1.2.2. Sección de tarjetas [1.5 ptos]

Esta sección debe estar dedicada a visualizar las tarjetas creadas para cada jugador todas las **tarjetas** de forma ordenada en formato de **grilla**. Idealmente debe permitir *scrolling* en caso de que no todas las tarjetas puedan estar en pantalla al mismo tiempo por su tamaño.

Esta sección también debe proveer una forma de **interacción** simple entre tarjetas: cada vez que se pase el cursor sobre una tarjeta de jugador, se deberán resaltar todas las tarjetas que tengan jugadores del **mismo equipo** que la interactuada. La forma en que se resalten dichas tarjetas quedará también a tu criterio, siempre y cuando sea diferenciable a simple vista.

1.2.3. Sección de resumen [1 pto]

Por otro lado, la segunda sección debe contener un **resumen de las tarjetas** visualizadas en la Sección de tarjetas e indicar la **cantidad total** de tarjetas mostradas. Además, debe mantenerse fija en una posición en la página, sin importar el *scrolling* de la Sección de tarjetas.

Como **resumen de las tarjetas** deberás implementar un **spider chart** que permita mostrar el promedio de las estadísticas de todas las cartas mostradas en la Sección de tarjetas. El **color** de polígono en este caso quedará a tu criterio, mientras no sea ninguno de los utilizados para codificar las cartas individuales.

1.3. Bonus: Filtros [0.4 pts]

En esta evaluación, puedes ir más allá e implementar más interacción en tu visualización. Esta consiste en agregar a la Sección de resumen una serie de *inputs* que permitirán filtrar las tarjetas que se visualizan en la Sección de tarjetas.

Hay dos tipos de filtros: **dropdown selects** y **range sliders**. En ambos casos, el resultado de filtración debe producir un cambio en ambas secciones de la página. Es decir, solo aquellas tarjetas que cumplan los filtros se muestran en la Sección de tarjetas, y la Sección de resumen solo debe considerar esas tarjetas para generar el resumen.

Las acciones a realizar mediante **dropdown selects** son el poder filtrar por valor de **liga** y por valor de **equipo**. Las opciones deben ser generadas a partir del *dataset*, es decir, si se cambia el *dataset* por uno con otras ligas, en el selector de ligas se deberán mostrar las ligas que tiene el nuevo *dataset*.

Las acciones a realizar mediante **range sliders** son el poder filtrar por valor de **RATING**. Para eso se especifica el rango de valores se usan los *inputs* mencionados para declarar el **RATING mínimo** y por **RATING máximo** del rango. No es necesario que implementes ambos en un solo *range*, puedes tener uno distinto para cada acción.

Por cada filtro de los anteriores (liga, equipo, RATING mínimo y RATING máximo) que implementes de forma correcta, obtendrás una bonificación de **0.1 puntos** (1 décima) a tu puntaje final.

2. Consideraciones

Al momento de implementar tu solución, deberás considerar lo siguiente:

- Tu programa puede cargar el *dataset* entregado directamente o cargar una versión procesada por ti. En el caso de hacer esto último, por favor entrega el código necesario para reproducir tu versión del *dataset*.

- Tu programa solo puede hacer uso de funciones nativas de JavaScript o provistas por D3.js, y no debe arrojar errores en la consola del navegador.
- Se espera utilices estilamiento nativo mediante CSS para esta evaluación, es decir, no se permite importar *frameworks* o herramientas de estilamiento ya construidas.
- Puedes hacer uso tanto de la versión 5 o 6 de D3.js, pero no una anterior a ambas.
- Eres libre de usar cualquier función provista por D3.js, y usar cualquier elemento SVG y sus características (como `rect`, `circle`, `path`, etc...).

3. Entregables

La entrega se realizará mediante la plataforma Canvas en la [evaluación correspondiente](#).

Para esta entrega se deberá subir un archivo de extensión **ZIP** que contenga un archivo de extensión HTML con todas las respuestas solicitadas anteriormente; y un programa de cargado de datos y generación de visualización JavaScript. Cualquier otro archivo (imágenes, hojas de estilo, *scripts*, etc.) que sea necesario para la visualización correcta de tu entrega debe también ir incluido en el archivo comprimido.

No se aceptarán entregas en cualquier otro formato distinto al indicado anteriormente (archivos PDF, TXT, DOC, etc...).

4. Rúbrica de evaluación

Se publicará en Canvas una planilla que contenga el criterio de evaluación en mayor detalle, con el fin de transparentar la forma en que se evaluará cada Hito.

5. Dudas

Cualquier duda que tengas sobre esta evaluación, prefiere publicarla en el [foro](#) correspondiente a esta evaluación. También siéntete libre de responder dudas de compañerxs si crees que manejas la respuesta.

6. Flexibilidad de entrega

En la eventualidad de que tengas problemas personales durante el plazo de esta evaluación, a tal punto que impida su realización de forma importante, siéntete libre de contactar a alguien del equipo docente para buscar apoyo y opciones de flexibilidad.

Es completamente posible otorgar una extensión de plazo individual. Se espera escribas con detalles de tu situación, al punto que sientas comodidad de hacerlo, para así entender y considerar tu caso.

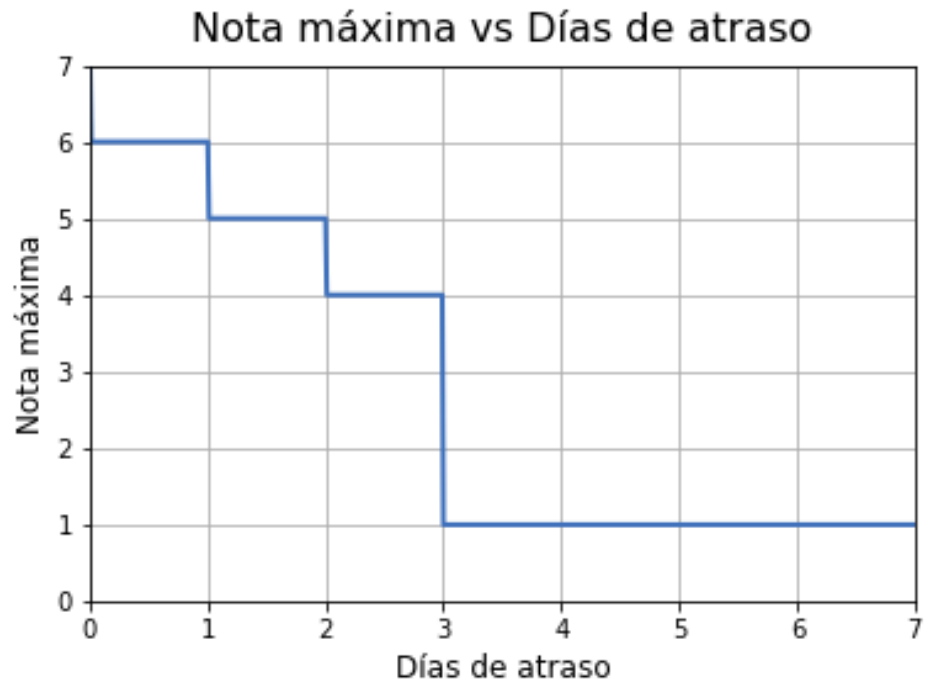
De preferencia escribe un correo al docente del curso (faflorenzano@ing.puc.cl) o a la ayudante jefa (diflores@uc.cl).

7. Política de atraso

Existe la posibilidad de entregar su hito con hasta **3 días de atraso** a partir de la fecha de entrega definida en el enunciado. En la eventualidad de entregar pasada la fecha de entrega, se aplicará una **reducción** a la nota máxima que podrá obtener en su hito.

De haber atraso, la nota máxima a obtener se reduce en **1 punto (10 décimas)** por cada día de atraso, siendo la nota final del hito calculada mediante la siguiente fórmula:

$$\text{mín}((7 - \text{días_atraso}), \text{nota_obtenida})$$



Mientras que cualquier hito que sea entregado con más de 3 días (72 hrs) de atraso será evaluado con la **calificación mínima (1.0)**.