



# Trabajo Integrador

Paradigmas de Programación 2024

---



## Pautas del trabajo

El trabajo será en grupos de MÁXIMO 4 integrantes.

Fecha de entrega: semana del 28/10/2024.

Para aprobar el trabajo:

- Se deberá realizar un diagrama de clases que acompañe la entrega.
- Se deberá utilizar un menú.
- Se deberá aplicar iteradores de colección.
- TODOS LOS INCISOS DEBEN FUNCIONAR, si hay al menos uno que no funcione se lo considerará desaprobado.

Este trabajo es condición para APROBAR LA CURSADA de Paradigmas de Programación.

## Inciso 1 - Sistema de Gestión de Cátedra

Un instituto de formación se encuentra en pleno crecimiento y necesita un sistema de gestión para sus cátedras. Se requiere desarrollar un software que permita gestionar a las personas dentro de cada cátedra, que pueden ser tanto estudiantes como docentes.

Se tiene que poder cargar la cátedra al sistema. De la misma se conoce código, nombre, carga horaria, carrera, docente a cargo, tipo (Curricular o Electiva) y nivel al que pertenece. Cada cátedra tiene un único docente.

A la cátedra se le debe poder agregar personas, de las cuales se guarda nombre, apellido, mail, legajo. Además, para los estudiantes se conoce la nota final y estado (Regular, Condicional, Libre, De Baja); y para los docentes el sueldo.

El cliente desea poder generar un listado de personas, así como poder buscarlas en base a su legajo ya sea para modificarlas o eliminarlas.

Al guardar las notas, solo se utilizarán notas numéricas para aquellos que hayan promocionado, de lo contrario se deberá registrar como "AC", para aquellos que aprobaron cursada, o "D" para los estudiantes que hayan desaprobado. Se debe poder mostrar un listado con los estudiantes aprobados, ya sea por cursada o por promoción. A su vez, guardar en un diccionario la cantidad de estudiantes por estado.

Como en muchas ocasiones los docentes van cargando los aprobados a medida que se avanza sobre las fechas de parcial, se debe poder filtrar aquellos alumnos sin nota y permitir actualizarlo.

Otro de los requerimientos del sistema es poder eliminar a todos los estudiantes que por diversos motivos hayan abandonado la cursada.

Finalmente, el sistema debe permitir calcular la nota promedio del curso y generar un listado de aquellos que superen la misma.

## Inciso 2 - Sistema de Gestión de Carrera

Un instituto de formación se encuentra en pleno crecimiento y necesita un sistema de gestión para sus carreras. Se requiere desarrollar un software que permita gestionar a las distintas cátedras dentro de cada carrera, que pueden ser anuales, cuatrimestrales o bimestrales.

El sistema debe permitir cargar la carrera con su nombre, cantidad de materias, cantidad de niveles y disciplina a la que pertenece.

La carrera cuenta con distinta cantidad de cátedras. De cada una se conoce código, nombre, carga horaria, legajo del docente a cargo, tipo (Curricular o Electiva) y nivel al que pertenece (cada carrera cuenta con cinco niveles). Además, si la materia es bimestral se debe conocer fecha de inicio y fecha de fin de cursada y si es cuatrimestral se debe conocer el cuatrimestre al que pertenece (Primero, Segundo o Ambos).

Se debe poder buscar la cátedra en base a su código para poder modificarla y/o eliminarla, así como generar un listado para ver todas las cátedras (esto también permitirá visualizar los cambios efectuados).

Cada cuatro años, el listado de materias electivas se renueva por lo que se deben poder mostrar y/o eliminar todas aquellas materias de este tipo y cargar nuevas (si así se desea).

Como al final del tercer nivel, la carrera ofrece un título intermedio, se debe poder mostrar un listado de cátedras excluyendo las de los últimos dos niveles.

Calcular la cantidad total de horas de la carrera. Guardar en un diccionario la cantidad de horas por tipo de cátedra (Anual, Cuatrimestral, Bimestral).

### Inciso 3 - Sistema de Gestión de Consultas Médicas

Un consultorio médico necesita un sistema de gestión para organizar a los pacientes que atiende diariamente. El software a desarrollar debe permitir:

El sistema debe permitir cargar los datos del consultorio, como el nombre del consultorio, la cantidad de pacientes diarios, el número de médicos disponibles y las especialidades que ofrece.

El consultorio atiende a una distinta cantidad de pacientes cada día. De cada consulta, se debe registrar número de consulta, DNI del paciente, nombre completo, edad, síntomas, médico asignado, fecha y hora de consulta (Se estima que las consultas se dan cada media hora). Además, se debe especificar si la consulta es de tipo regular o de urgencia.

Para las consultas regulares, se debe almacenar también el diagnóstico y la receta si corresponde. En el caso de las consultas de urgencia, se debe registrar la prioridad (Alta, Media o Baja) y el tiempo de atención estimado.

El sistema debe permitir buscar a una consulta específica mediante número de consulta para modificar su información o eliminar su registro. Tener en cuenta que las consultas regulares no pueden superponerse. Si se diera el caso en que entra una consulta de urgencia en el horario de una regular, se debe retrasar la consulta regular (y sus consultas posteriores, si las hubiese) media hora.

Se debe poder generar un listado con todas las consultas realizadas o por realizar en un día determinado, mostrando tanto las regulares como las de urgencia.

Se debe poder mostrar una lista solo con las consultas urgentes, ordenadas por prioridad (de Alta a Baja).

Muchas veces es necesario realizar un informe de la edad promedio de los pacientes que frecuentan distintas franjas horarias. Se debe poder elegir la franja a mostrar, ya sea mañana (de 8 a 13), tarde (13 a 18) y noche (18 a 23).

Al irse de vacaciones, los médicos (usuarios del sistema), podrían querer ver las consultas que hayan reservadas para ir acomodando sus agendas, por lo que se debe permitir generar un listado de consultas, excluyendo el plazo de vacación del doctor.

Generar un diccionario que guarde la cantidad de consultas por paciente.

## Inciso 4 - Sistema de Gestión de Hospital

Un hospital de La Plata desea gestionar todos sus consultorios. El software debe permitir cargar nuevos consultorios al hospital. De cada uno se conoce el número de consultorio (no debe repetirse), piso (Planta baja, primer piso, segundo piso, etc), ala (norte, sur, este u oeste), número de matrícula del médico que lo atiende.

No solo existen los consultorios generales, sino que el hospital cuenta también con consultorios específicos que tienen un horario de atención específico y la especialidad que ocupan; y se tiene de urgencias de los que se desea conocer el estado ya que al no tener cita previa, se debe permitir al usuario consultar si está ocupado o no.

Se debe poder buscar el consultorio por número. Esto podrá servir también para poder modificar los datos o eliminar un consultorio específico.

Dado que los consultorios específicos no atienden los fines de semana, el usuario debe poder ver un listado que no contemple a los mismos.

A modo de guía, el sistema debe proveer un listado de consultorios por planta. Por otro lado, se desea poder generar un diccionario que guarde la cantidad de consultorios por planta.

Por último, debe permitir informar el porcentaje de consultorios de urgencia ocupados y el porcentaje libre.

## Inciso 5 - Desarrollo de juego de carreras

Se encuentran en un estudio de desarrollo de videojuegos. Se les asigna el desarrollo del motor de un juego arcade de carreras de autos. En el juego, el jugador tiene diferentes autos, los cuales pueden ser personalizados por el jugador (color y patente). Además, se tiene en cuenta la marca y modelo; también, las estadísticas de aceleración, freno y dirección. Las diferentes marcas tienen un potenciador diferente afectando a las estadísticas del auto, Audi tiene un boost de velocidad, Volkswagen puede incorporar nuevo equipamiento y Ford pueden llegar a tener nitro. El sistema debe cargar los autos del jugador.

La idea del juego es que cuanto más avance el jugador consigue más dinero, con el cual consigue los autos. Pero para la fase temprana del desarrollo, solo se cargarán los autos al garage del jugador si desde el sistema lo hace el diseñador, que cargará color, patente, aceleración, freno, dirección y marca, siendo esta última característica una serie de opciones que ya están designadas desde el motor de juego (su sistema).

El sistema debe permitir la cargar nuevos autos, así como buscarlos por patente. Este filtro permitirá acceder más rápido a sus características para poder modificarlos o eliminarlos. Tener en cuenta que al mostrar las estadísticas afectan los modificar.

El usuario debe poder elegir el auto que desee, y muchas veces optan por filtrar en base al color, por lo que se debe poder generar un listado de autos de color determinado mostrando su marca.

Se debe poder diseñar un nivel que deba eliminar los autos de una determinada marca.

Se debe poder diseñar un modo que decremente la dirección del primer auto que se encuentre de un color dado.

Se debe poder diseñar un modo que incremente la dirección del primer auto que no cumpla con un color dado.

## Inciso 6 - Gestión de juegos de consola

Se requiere desarrollar un sistema que simule una Consola de Videojuegos. La consola debe ser capaz de gestionar distintos juegos de los que se conoce id del juego, título, género, peso, plataformas compatibles y consumo de ram. A su vez, debe comprender los que sean SinglePlayer con su duración de campaña y los MultiPlayer con el máximo de jugadores y si es online o no. Se debe tener cuidado de que los juegos que se cargan NO EXCEDAN la capacidad total de almacenamiento de la consola.

El sistema debe permitir cargar tantos juegos como desee a la consola, así como buscar por id. Esto va a permitir filtrarlos más rápido, así como recuperarlos para poder modificarlos y eliminarlos.

A su vez, el usuario desea poder consultar el consumo de RAM promedio de todos los juegos.

Al emular una consola, muchas veces se necesita conocer el listado de juegos disponibles, ya que dependiendo de la plataforma es lo que se va a permitir jugar, por lo que el usuario debe poder filtrar en base a su plataforma de preferencia.

Ante distintas eventualidades, puede que se dé el caso donde los usuarios no puedan jugar online, por lo que se debe poder generar el listado de juegos que no tengan esta modalidad.

Finalmente, el sistema debe permitir generar un diccionario con la cantidad de juegos de los que se dispone clasificados por género.



## Inciso 7 - Sistema de Gestión de torneo de fútbol.

Contexto: Una liga semiprofesional del deporte "fútbol" local, se encuentra en pleno auge, y necesita un sistema que les sea de ayuda. Están en búsqueda de un software mediante el cual se gestione a la liga, y a los clubes de la misma.

Se debe poder cargar la liga al sistema. Los datos que interesan son el nombre, el país, la categoría (1, 2, o 3), código de identificación, y premio al primer lugar.

A la liga se le deben poder AGREGAR CLUBES, de los cuales se guarda nombre, año de creación, puesto final, capacidad de espectadores, y barrio donde se encuentra la cancha. Si el club ya ha salido campeón alguna vez, interesa guardar el año del último campeonato, cuantos goles metió en esta campaña y el color principal de su camiseta titular.

El cliente debe poder generar un listado de equipos, buscarlos por puesto, así como modificarlos o eliminarlos de la liga.

Se debe poder mostrar un listado de clubes con un año de creación menor al año 2000.

Filtrar aquellos clubes con una capacidad de espectadores menor a 10000 personas.

Un requerimiento del sistema es que se debe poder eliminar a aquellos cuyo puesto es 19 o 20.

El sistema debe permitir calcular un promedio de capacidad de espectadores, y mostrar a aquellos que estén por encima del doble del promedio, para saber qué equipos son aquellos que más entradas pueden vender.

El cliente busca mostrar en su página al club campeón que más goles haya marcado en su campaña consagrada, elaborar una funcionalidad que le permita saber al cliente que club es el indicado y cuantos goles son los que anotó.

Se debe poder guardar la cantidad de clubes por barrio en un diccionario.

## Inciso 8 - Sistema de Gestión de Club de fútbol.

Un club en particular de la Liga Amateur Platense está en pleno crecimiento y se encuentra en búsqueda de un sistema para digitalizar la gestión del club, el cuerpo técnico, y los jugadores.

Se debe permitir la carga del club al sistema, con su nombre, nombre del estadio, año de creación, cantidad de campeonatos, y división (1,2 o 3).

Del plantel interesa que se puedan cargar jugadores con su nombre, apellido, número de camiseta (único), y fecha de finalización de contrato.

De los arqueros y defensores, interesa saber sus vallas invictas en el club (cantidad de partidos sin recibir un gol), y tanto de los mediocampistas como de los delanteros, la cantidad de goles y la cantidad de asistencias de cada uno.

El cliente debe poder generar un listado de todos los jugadores del plantel, así como buscarlos por número de camiseta para modificar datos como la fecha de finalización de contrato o eliminarlos si se rompe la relación con el club.

El sistema debe permitir una búsqueda del jugador que utiliza la camiseta "10" e imprimir su nombre y apellido en pantalla.

El usuario necesita que el sistema le permita imprimir un listado de todos aquellos atacantes que no hayan convertido ningún gol para analizar su rendimiento.

El sistema debe permitir eliminar a todos aquellos jugadores que se hayan pasado de la fecha de finalización del contrato.

Es de interés del usuario, que el sistema pueda mostrar un promedio de las vallas invictas de los jugadores defensivos (arqueros y defensores).

El sistema debe tener una funcionalidad que permite calcular los días restantes de contrato.

Se debe poder guardar la cantidad de atacantes según la cantidad de goles anotados en un diccionario y mostrarlo en pantalla.