

Apellido y Nombre: Legajo:

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección.

EquiApp - Buscando al Equipo Perfecto

EquiApp tiene como objetivo principal facilitar la formación de equipos ideales para la realización de trabajos prácticos dentro de la carrera de Ingeniería en Sistemas de Información de la UTN BA.

EquiApp busca revolucionar la manera en que los estudiantes conforman equipos para proyectos académicos, centrándose en la optimización de habilidades y aptitudes de los integrantes. El sistema utilizará el historial académico de los estudiantes como base para la formación de equipos, teniendo en cuenta variables como el desempeño en asignaturas previas, competencias técnicas, y habilidades interpersonales.

Datos básicos de Estudiantes

Todo estudiante que quiera utilizar el Sistema deberá ingresar su número de legajo, el cual es único dentro de toda la regional. En base a este legajo, el Sistema buscará, en primer lugar, los datos personales en el *Sistema de Gestión de Estudiantes* y luego, el historial académico en el *Sistema de Gestión Académica*, en ambos casos mediante una API REST.

Ver en la sección “Ejemplos de Solicitudes y Respuestas” ejemplos de las solicitudes y respuestas de ambos Sistemas.

Cálculo de Desempeño

Luego de haber ingresado el legajo y que el Sistema haya buscado los datos anteriormente mencionados, EquiApp deberá calcular y guardar registro de los “*datos de desempeño*” de un estudiante. Los datos de desempeño son los siguientes:

- cantidad de asignaturas que cursó más de una vez;
- cantidad de asignaturas firmadas actuales;
- cantidad de asignaturas aprobadas (por final);
- cantidad de asignaturas promocionadas;
- promedio general.

Para buscar las notas de las cursadas, el Sistema de Gestión Académica expone otra API REST que permite consultar el detalle de una cursada, brindando las calificaciones obtenidas por el estudiante en la misma (1 calificación por cada parcial).

Cuestionario de Habilidades Interpersonales y Habilidades Técnicas

Por último, la persona deberá completar, para terminar con el proceso de registro, un cuestionario de “Habilidades interpersonales” y otro de “Habilidades técnicas” según se detalla en la siguiente sección.

El Sistema deberá permitir la gestión del cuestionario de Habilidades Interpersonales y del cuestionario de Habilidades Técnicas a los Administradores de la plataforma, para que los mismos sean completados por los estudiantes al momento de registro. Estos cuestionarios servirán para poder generar una “buena combinación” de personas para la formación de equipos.

Las preguntas al cuestionario serán configuradas y administradas (no se encuentran *hardcodeadas*) y deberán poder soportar preguntas de respuesta múltiple con 1 sólo respuesta posible, por ejemplo:

- ¿Cómo te adaptas a diferentes estilos de trabajo y personalidades dentro de un equipo? Seleccione todas las opciones que corresponda:
 - Opción 1
 - Opción 2
 - Opción 3

Se prevé en el futuro incorporar preguntas que aceptan más de 1 respuesta y preguntas de respuesta abierta. Tener en cuenta al momento de proponer una solución.

Búsqueda de equipos

Una vez dentro de la plataforma, los alumnos podrán dar de alta una nueva “búsqueda de equipo”. Para esto deberán especificar:

- Plan de la carrera: el Sistema deberá mostrar un desplegable con todos los planes vigentes de la carrera. Cada plan posee un código, su descripción y el año de puesta en marcha.
- Asignatura: el Sistema deberá mostrar un desplegable con todas las asignaturas del plan seleccionado. De cada asignatura se conoce su código y su nombre.
- Cantidad de integrantes mínima y cantidad de integrantes máxima
- Descripción de la búsqueda

Los planes y las asignaturas podrán ser obtenidos llamando al Sistema de Gestión Académica, el cual expone dos *endpoints* REST para este motivo.

Algoritmo de Coincidencias

Para esta primera etapa, será el Sistema únicamente quien sugiera las posibles coincidencias para el armado de los equipos de forma asincrónica; quedando por fuera del alcance la postulación “manual” de un alumno a una búsqueda abierta por otro estudiante.

El algoritmo de coincidencias está siendo desarrollado, en paralelo, por otro equipo. Nos han comentado que expondrá una interfaz REST a la cual podremos llamar para que genere los posibles equipos dada una búsqueda. Este Sistema implementará el mecanismo de WebHook¹, ya que para generar los posibles equipos necesita “venir a consultar a nuestro componente” por datos de los estudiantes y sus historiales académicos. En la solicitud debemos enviarle el id de la búsqueda y una dirección de callback, que llamará al finalizar la generación de posibles equipos.

Resultado de búsquedas de equipos

Una vez que el componente que implementa el Algoritmo de Coincidencias finalice su ejecución y llame a nuestra dirección de Callback establecida, el Sistema debe asociar los resultados obtenidos con la búsqueda. En ese momento, el Sistema deberá marcar la búsqueda como “Sugerida” y notificarle al estudiante que ya se encuentran disponibles las sugerencias.

Ver en la sección “Ejemplos de Solicitudes y Respuestas” ejemplo del resultado arrojado por el componente de algoritmo de coincidencias.

¹ Los webhooks son un mecanismo de comunicación que también se conocen como API inversas o API push porque, al usarlos, quien debe encargarse de la comunicación es el servidor, en vez del cliente. Es decir, el servidor le envía al cliente una solicitud POST única de HTTP cuando los datos están disponibles, en lugar de que el cliente le envíe solicitudes de HTTP hasta que responda.



Aceptación de conformación

Cuando el estudiante observa los resultados de la búsqueda de equipos podrá marcar la aceptación o no de esa conformación. Y cuando el 80% de los integrantes del equipo propuesto realizan la aceptación, el equipo pasa a estado "Aceptado".

Alcance y requerimientos

El Sistema deberá permitir:

- El registro de los estudiantes, obteniendo sus datos a partir del Legajo.
- Calcular y guardar los "datos de desempeño" de los estudiantes.
- Gestión del cuestionario de Habilidades Interpersonales por parte de los administradores.
- Gestión del cuestionario de Habilidades Técnicas por parte de los administradores.
- Alta de Búsqueda de Equipos por parte de los estudiantes.
- Visualización de las sugerencias generadas por el Algoritmo de Coincidencias, a los estudiantes
- Aceptación de la sugerencia de equipos

Queda por fuera del alcance:

- La notificación que el Sistema debe enviarle al estudiante una vez que la búsqueda entra en estado "Sugerida".
- La postulación "manual" de un estudiante a una búsqueda abierta por otro estudiante.



Ejemplos de Solicitudes y Respuestas

- Request a Sistema de Gestión de Estudiantes

- Solicitud: GET /api/estudiante/199-999-9
- Respuesta:

```
{
  "legajo": "199-999-9",
  "nombre": "Román",
  "apellido": "Pérez",
  "fecha_nacimiento": "23/10/2005",
  "direccion": "Calle Falsa 123",
  "barrio": "Almagro",
  "fecha_ingreso": "10/12/2018",
  "id_interno": "d341d77b-b33a-419a-9eae-bb52752dfde7"
}
```

- Request a Sistema de Gestión Académica

- Solicitud: GET /api/historial-academico/d341d77b-b33a-419a-9eae-bb52752dfde7
- Respuesta:

```
[
  {
    "asignatura_id": "73c5994e-82fd-4caf-a76e-9d4a2bda8bdb",
    "estado": "FIRMADA",
    "fecha_ult_estado": "10/10/2021",
    "cant_veces_cursada": 2,
    "cursada_id": "c7dc6bb7-946c-46bd-ba1c-c4a1d8f280ab"
  },
  {
    "asignatura_id": "98c5994e-82fd-4caf-a76e-9d4a2bda8bdb",
    "estado": "PROMOCIONADA",
    "fecha_ult_estado": "10/12/2018",
    "cant_veces_cursada": 1,
    "cursada_id": null
  },
  {
    "asignatura_id": "45c5994e-82fd-4caf-a76e-9d4a2bda8bdb",
    "estado": "PENDIENTE",
    "fecha_ult_estado": "10/12/2018",
    "cant_veces_cursada": 0,
    "cursada_id": null
  }
]
```

- Algoritmo de coincidencias

```
{
  "busqueda_id": "a5a3dc16-bb57-4d32-84a3-20b08e21a6c3",
  "fecha_generacion": "16/12/2023",
  "resultados": [
    {
      "id_resultado": "efdaa412-36d2-4cfe-bbdf-5826aac9bc6f",
      "estudiantes": [
        "d3411930-4cfc-47a9-ae31-c21a6bba0c73",
        "5ff763f2-3486-4e64-b9a4-22b703015578",
        "87d00f8c-a970-4378-a0ca-04d33d526660"
      ]
    }
  ]
}
```



Punto 1 – Arquitectura (35 puntos)

1. **(10 puntos)** Comunique mediante un Diagrama de Componentes y un Diagrama de Despliegue de UML la arquitectura del Sistema, indicando componentes, nodos, relaciones, protocolos y todo lo que considere necesario.
2. **(10 puntos)** Enumere los recursos que necesita exponer nuestro Sistema para que el Componentes “Algoritmo de coincidencias” pueda funcionar. Además, establezca al menos dos endpoints completos de nuestro Componente, especificando claramente por cada uno: verbo, ruta, cuerpo y respuesta.
3. **(15 puntos)** Suponiendo que nos interesa utilizar un componente externo para el envío de notificaciones al estudiante, y sabiendo que éste componente puede demorar “más de lo esperado” en ciertas ocasiones cuando lo llamamos; ¿cómo haría para que el componente de Algoritmo de Coincidencias no cierre la conexión por Time Out cuando nos envía los resultados, a nuestra dirección de Callback establecida, y se le tiene que notificar al estudiante sobre éstos? Brinde un detalle completo.

Punto 2 - Modelo de Dominio (35 puntos)

- A. **(20 Puntos)** Documentar la solución utilizando diagramas UML (diagrama de clases obligatorio).
- B. **(15 Puntos)** Justificar las decisiones de diseño que se tomen, por ejemplo, haciendo referencia a los principios que guían al diseño o las consecuencias de aplicar un determinado patrón. También puede optar por justificar mediante código, pseudocódigo o algún otro diagrama complementario.

Punto 3 – Persistencia (30 puntos)

- A. **(20 Puntos)** Diseñar el modelo de datos del punto anterior para poder persistir en una base de datos relacional, indicando las entidades con sus respectivos campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda.
- B. **(15 Puntos)** Justificar:
 - Qué elementos del modelo es necesario persistir.
 - Cómo resolvió los *impedance mismatches*.
 - Las estructuras de datos que deban ser desnormalizadas, si corresponde.