



**UNIVERSIDAD DE SONORA
CAMPUS HERMOSILLO**

ING. EN SISTEMAS DE INFORMACION

DOCUMENTACION: Base de Datos

➤ **Responsables:**

- **Líder:**
 - Fontes Fernández Alejandro
- **Integrantes:**
 - Contreras Duarte Mariangel Lilibeth
 - Aragón Romero Diana Carolina
 - Celaya García Juan Carlos
 - Zavala Rocha José Ignacio
 - Betancourt Laurean Tania Berenice
 - Silva Fragoso Jonatan Eduardo

TABLAS UTILIZADAS

Nombre	Campos	Objetivo
alumno	<ul style="list-style-type: none"> • id • expediente • nombre • depto (departamento) • camus • prog (programa) • opcion • especialidad • plan • status (estatus) • uciclo (último ciclo) • uinsc (ultima inscripción) • riesgo • riesgoant (riesgo anterior) 	En esta tabla podemos conseguir la información de los alumnos de la universidad.
alum_acad	<ul style="list-style-type: none"> • id • expediente • depto (departamento) • programa • plan • cambios • cred (créditos) • status (estatus) • tipo • cred_apro (créditos aprobados) • prom_sem (promedio semestral) • prom_per (promedio periodo) • inicio (inicio de la carrera) • ultimo (último periodo en la carrera) • periodo (periodo actual) • pro (periodos completados) • mats (materias) • matakum (materias acumuladas) 	En esta tabla podemos ver la información de los alumnos en lo que tiene que ver con las materias.
departamento	<ul style="list-style-type: none"> • id • clave • division • campus • Descripcion 	Esta tabla nos da la información de los departamentos de la universidad.
inscripcion	<ul style="list-style-type: none"> • id • expediente • depa (departamento) • prog (programa) • clave • creditos • grupo 	En esta tabla se puede obtener la información de todas las inscripciones de los alumnos.

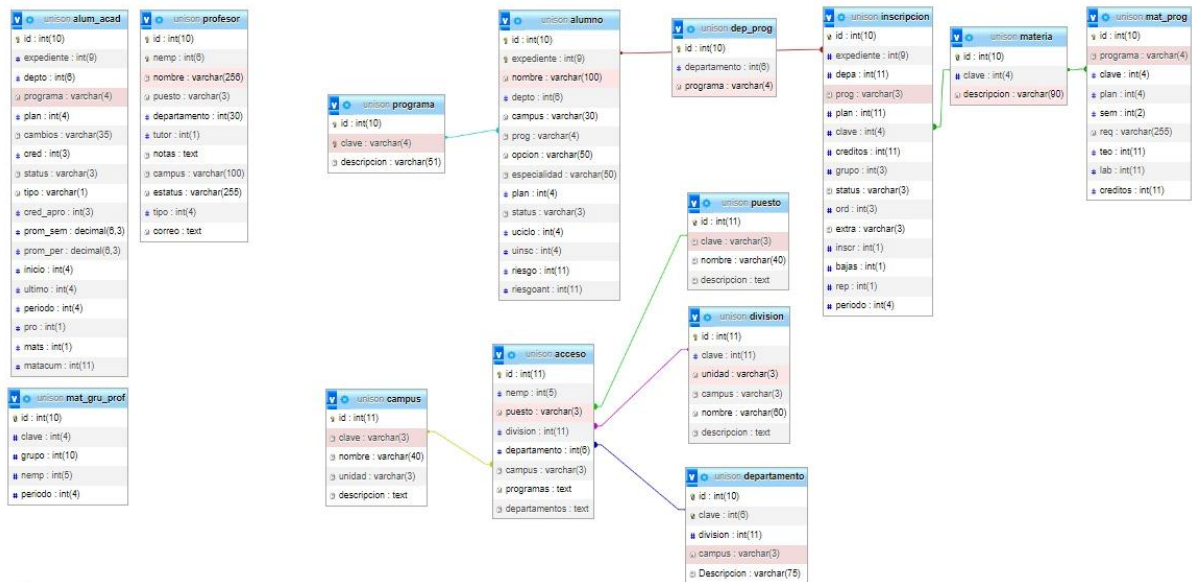
	<ul style="list-style-type: none"> • status (estatus) • ord (calificación ordinaria) • extra (calificación extraordinaria) • inscr (veces inscritas) • bajas (cantidad de bajas) • rep (cantidad de reprobadas) • periodo 	
materia	<ul style="list-style-type: none"> • id • clave • descripcion 	En esta tabla se presentan todas las materias impartidas en la universidad.
mat_gru_prof	<ul style="list-style-type: none"> • id • clave • grupo • nemp (número de empleado) • periodo 	En esta tabla podemos ver los profesores que imparten una materia y a que grupos la imparte.
mat_prog	<ul style="list-style-type: none"> • id • programa • clave • plan • sem (semestre) • req (requisitos) • teo (horas de teoría) • lab (horas de laboratorio) • credits 	En esta tabla podemos ver la información de las materias como lo es el plan, semestre, horas teóricas y de laboratorio, así como los créditos.
profesor	<ul style="list-style-type: none"> • id • nemp (número de empleado) • nombre • puesto • departamento • tutor • notas • campus • estatus • tipo • correo 	En esta Tabla podemos obtener la información de los profesores de la universidad.
programa	<ul style="list-style-type: none"> • id • clave • descripcion 	En esta tabla están todas las carreras ofrecidas en la universidad.
acceso	<ul style="list-style-type: none"> • id • nemp (número de empleado) • puesto • division • departamento • campus • programas • departamentos 	En esta tabla podemos obtener la información de los empleados de la universidad.

campus	<ul style="list-style-type: none"> • id • clave • nombre • unidad • descripcion 	Esta tabla nos enlista todos los campus que tiene la universidad de sonora.
dep_prog	<ul style="list-style-type: none"> • id • departamento • programa 	Esta tabla nos dice cuáles son los programas de cada departamento.
division	<ul style="list-style-type: none"> • id • clave • unidad • campus • nombre • descripcion 	Esta tabla nos muestra la información de las divisiones de la universidad.
puesto	<ul style="list-style-type: none"> • id • clave • nombre • descripcion 	Esta tabla nos da la información de los diferentes puestos que maneja la universidad.

RELACION

Nombre	Descripción
alumnos	<ul style="list-style-type: none"> • inscripción: Se relacionan mediante el campo expediente, que se ha llamado así en ambas tablas, por lo que a cada expediente se le relaciona con uno o varios registros de la tabla inscripción. • programa: Se relacionan mediante el campo clave (tabla programa), y el campo prog (tabla alumno), por lo que ha cada campo clave se le relaciona con uno o más registros de la tabla alumnos.
departamento	<ul style="list-style-type: none"> • acceso: Se relacionan mediante el campo departamento (tabla acceso) y el campo clave (tabla departamento), por lo que a cada campo departamento se le relaciona solo con un campo clave.
inscripcion	<ul style="list-style-type: none"> • alumno: Se relacionan mediante el campo expediente, que se ha llamado así en ambas tablas, por lo que a cada expediente se le relaciona con uno o varios registros de la tabla inscripción. • materia: Se relacionan mediante el campo clave, que ha sido llamado así en ambas tablas, por lo que cada clave de materia se relaciona con uno o más registros de la tabla inscripción.
materia	<ul style="list-style-type: none"> • mat_prog: Se relacionan mediante el campo clave, llamado así en ambas tablas, por lo que cada clave de materia se relaciona con uno o más registros de la tabla mat_prog. • inscripcion: Se relacionan mediante el campo clave, que ha sido llamado así en ambas tablas, por lo que cada clave de materia se relaciona con uno o más registros de la tabla inscripción.
mat_prog	<ul style="list-style-type: none"> • materia: Se relacionan mediante el campo clave, llamado así en ambas tablas, por lo que cada clave de materia se relaciona con uno o más registros de la tabla mat_prog.
programa	<ul style="list-style-type: none"> • alumno: Se relacionan mediante el campo clave (tabla programa), y el campo prog (tabla alumno), por lo que ha cada campo clave se le relaciona con uno o más registros de la tabla alumnos.
acceso	<ul style="list-style-type: none"> • puesto: Se relacionan mediante el campo puesto (tabla acceso), y el campo clave (tabla 'puesto'), por lo que, cada registro del campo clave se relaciona con uno o más registros de la tabla acceso. • divison: Se relacionan mediante el campo división (tabla acceso), y el campo clave (tabla división), por lo que, cada registro del campo clave se relaciona con uno o más registros de la tabla acceso. • departamento: Se relacionan mediante el campo departamento (tabla acceso) y el campo clave (tabla departamento), por lo que, cada campo departamento se relaciona con uno o más registros de la tabla departamento. • campus: Se relacionan mediante el campo clave (tabla campus) y el campo campus (tabla acceso), por lo que, cada registro del campo clave se relaciona con uno o más registros de la tabla acceso.

campus	<ul style="list-style-type: none"> acceso: Se relacionan mediante el campo clave (tabla campus) y el campo campus (tabla acceso), por lo que, cada registro del campo clave se relaciona con uno o más registros de la tabla acceso.
division	<ul style="list-style-type: none"> acceso: Se relacionan mediante el campo división (tabla acceso), y el campo clave (tabla división), por lo que, cada registro del campo clave se relaciona con uno o más registros de la tabla acceso.
puesto	<ul style="list-style-type: none"> acceso: Se relacionan mediante el campo puesto (tabla “acceso”), y el campo clave (tabla “puesto”), por lo que, cada registro del campo clave se relaciona con uno o más registros de la tabla acceso.



QUERYS

PRINCIPALINFORMATION

Query	Descripción
SELECT a.nombre, a.expediente, p.descripcion, a.prog, a.campus, (CASE WHEN a.status != 'E' AND a.status != 'I' AND a.status != 'I40' AND a.status != 'S' THEN 'ACTIVO' ELSE 'No Activo' END) AS estatus, (CASE WHEN aa.tipo = 'R' THEN 'REGULAR' ELSE 'IRREGULAR' END) as tipoAlumno FROM alumno a, alum_acad aa, programa p WHERE a.expediente = 221204595 AND a.expediente = aa.expediente AND p.clave = a.prog	La query trae el nombre, expediente, el nombre del programa, las siglas del programa, el campus, el estatus y el tipo de alumno que sea el alumno, esto para la parte del perfil del alumno.
SELECT a.nombre, aa.status FROM alumno a, alum_acad aa WHERE a.expediente=aa.expediente AND a.expediente = ? AND (aa.status = 'A' OR aa.status = 'B38');	Esta query nos da el nombre del alumno, así como su estatus.
SELECT profesor.nombre FROM profesor, acceso WHERE profesor.nemp = acceso.nemp AND acceso.nemp = ? AND acceso.nemp = ? AND profesor.estatus = 'ACTIVO' AND acceso.puesto = 'CP';	Esta query verifica si es un coordinador de programa y está en activo para luego dar el nombre y pedir su número de empleado y su acceso.
SELECT nombre, prog, campus, plan FROM alumno WHERE expediente= ?;	Esta query nos trae el nombre del alumno, el programa en el que está, el campus y el plan al que pertenece.
SELECT descripcion FROM programa WHERE clave='?';	Esta query nos trae la descripción del programa dado.
SELECT COUNT(DISTINCT i.periodo) AS semestre FROM inscripción i, alum_acad aa WHERE i.expediente=? AND i.prog = aa.programa;	Nos dice el semestre que está cursando.
SELECT cred, cred_apro FROM `alum_acad` WHERE expediente = ?;	Nos dice los créditos aprobados y créditos que necesita el alumno.
SELECT alumno.depto, departamento.Descripcion FROM alumno, departamento WHERE alumno.depto = departamento.clave AND alumno.expediente = ? LIMIT 1;	Nos dice a la clave y nombre del departamento al que pertenece el alumno.

SELECT status,tipo FROM alum_acad WHERE expediente = ?;	Nos dice el estatus del alumno, así como el tipo de alumno.
SELECT ultimo FROM alum_acad WHERE expediente = ?;	Nos trae el último periodo de inscripción del alumno.
SELECT nombre, campus, correo, estatus FROM profesor WHERE nemp = ?;	Nos trae el nombre, campus, correo y estatus del coordinador.
SELECT puesto, division, departamento, programas FROM acceso WHERE nemp= ?;	Nos trae el puesto, división, departamentos y programas pertenecientes al coordinador.
SELECT descripcion FROM programa WHERE clave = ?;	Nos trae el nombre del programa del coordinador.
SELECT nombre FROM division WHERE clave= ?;	Trae el nombre de la división perteneciente al programa del coordinador.
SELECT Descripcion FROM departamento WHERE clave= ?;	Nos dice el nombre del departamento del coordinador.
SELECT nombre FROM puesto WHERE clave = ?;	Nos dice cuál es el puesto del coordinador.

QUERYS

ALUMNOS

Query	Descripción
SELECT clave FROM inscripcion WHERE expediente = (?) AND status = 'A';	En base al expediente del alumno, se solicitan las claves de las materias que el alumno ha aprobado.
SELECT clave FROM inscripcion WHERE expediente = (?) AND status = 'C';	En base al expediente del alumno, se solicitan las claves de las materias que el alumno se encuentra cursando.
CREATE TEMPORARY TABLE IF NOT EXISTS materiasAlumnoTemporal AS SELECT m.descripcion, m.clave, 'PC' AS estado, i.prog, i.plan FROM materia m, mat_prog mp, (SELECT * FROM inscripcion i3 WHERE i3.expediente =(?)) AS i WHERE mp.sem <=(?) AND m.clave = mp.clave AND NOT EXISTS(SELECT i2.clave, i2.status FROM inscripcion i2 WHERE i2.expediente =(?) AND mp.clave = i2.clave) AND mp.plan = i.plan AND mp.programa = i.prog AND mp.clave != 119 AND mp.clave != 660 AND mp.clave != 733 AND mp.clave	Se crea una tabla temporal donde se ingresan la descripción, clave, programa y plan de las materias que no han sido cursadas por el alumno o han sido reprobadas o dadas de baja, además de las materias que se le abrirán el siguiente semestre correspondiente.

```

!= 19006 AND mp.clave != 81 AND m.descripcion != 'PRÁCTICAS
PROFESIONALES'
GROUP BY
    m.descripcion
UNION
SELECT
    m.descripcion,
    m.clave,
    (
        CASE WHEN(i4.clave = m.clave) THEN i4.status ELSE 'PC'
        END
    ),
    i.prog,
    i.plan
FROM
    materia m,
    mat_prog mp,
    (
        SELECT
            *
        FROM
            inscripcion i3
        WHERE
            i3.expediente =(?)
    ) AS i,
    (
        SELECT
            i3.clave,
            i3.status
        FROM
            inscripcion i3
        WHERE
            i3.expediente =(?)
    ) AS i4
WHERE
    mp.sem <=(¿ ?) AND m.clave = mp.clave AND i4.clave =
mp.clave AND NOT EXISTS(
    SELECT
        i2.clave,
        i2.status
    FROM
        inscripcion i2
    WHERE
        i2.expediente =(?) AND(i2.status = 'A' OR i2.status = 'C') AND
mp.clave = i2.clave
) AND mp.plan = i.plan AND mp.programa = i.prog AND mp.clave
!= 119 AND mp.clave != 660 AND mp.clave != 733 AND mp.clave
!= 19006 AND mp.clave != 81 AND m.descripcion != 'PRÁCTICAS
PROFESIONALES'
GROUP BY
    m.descripcion;

```

```

SELECT
    materia.descripcion AS descripcion,
    ROUND(
        SUM(

```

Se muestran las materias de la tabla anterior con la siguiente información:

```

CASE WHEN(
    inscripcion.ord > 0 OR inscripcion.ord IS NOT NULL
) AND inscripcion.plan = ¿ ? AND inscripcion.prog = ?
THEN inscripcion.ord ELSE(
    CASE WHEN(
        inscripcion.extra > 0 OR inscripcion.extra IS NOT
NULL
    ) AND inscripcion.plan = ¿ ? AND inscripcion.prog = ?
THEN inscripcion.extra
    END
)
) / SUM(
    CASE WHEN(
        inscripcion.bajas = 0 OR inscripcion.bajas IS NULL
    ) AND inscripcion.plan = ¿ ? AND inscripcion.prog = ? THEN 1
ELSE 0
END
),
2
) AS promedioMateria,
ROUND(
    (
        SUM(
            CASE WHEN(
                inscripcion.bajas > 0 AND inscripcion.bajas IS NOT
NULL
            ) AND inscripcion.plan = ¿ ? AND inscripcion.prog = ?
THEN 1 ELSE 0
            END
        ) / COUNT(
            CASE WHEN(
                inscripcion.bajas >= 0 OR inscripcion.bajas IS NOT NULL
            ) AND inscripcion.plan = ¿ ? AND inscripcion.prog = ? THEN
1 ELSE 0
            END
        )
    ) * 100,
2
) AS indiceBajas,
ROUND(
    (
        SUM(
            CASE WHEN(
                inscripcion.ord >= 60 OR inscripcion.extra >= 60
            ) AND inscripcion.status = 'A' AND inscripcion.plan = ¿ ?
AND inscripcion.prog = ? THEN 1 ELSE 0
            END
        ) / SUM(
            CASE WHEN(
                inscripcion.status = 'A' OR inscripcion.status = 'R'
            ) AND inscripcion.plan = ¿ ? AND inscripcion.prog =
programa THEN 1 ELSE 0
            END
        )
    )
)

```

- Nombre de la materia.
- Promedio de la materia.
- Índice de bajas de la materia.
- Porcentaje de aprobación de la materia.
- Cantidad de bajas.
- Cantidad de alumnos inscritos.
- Estado para el alumno (reprobada, baja voluntaria o por cursar).
- Requisitos para abrir la materia.
- Semestre en que se abre la materia.
- Número de veces inscritas o intentos que ha tenido en la materia el alumno.

Estos datos varían según el programa del alumno, por ejemplo, no habrá las mismas bajas en cálculo 1 en Sistemas que en Industrial.

<pre>) AS req, mat_prog.sem as semestre, COALESCE(inscripcion2.inscr, 0) AS intentos FROM inscripcion JOIN materia ON inscripcion.clave = materia.clave JOIN mat_prog ON materia.clave = mat_prog.clave JOIN materiasAlumnoTemporal ON materia.clave = materiasAlumnoTemporal.clave LEFT JOIN inscripcion AS inscripcion2 ON inscripcion2.clave = mat_prog.clave AND inscripcion2.expediente = (?) WHERE mat_prog.programa = ? AND mat_prog.plan = ¿ ? GROUP BY materia.descripcion, mat_prog.req, materia.clave; </pre>	
<pre> SELECT DISTINCT (materia.descripcion), inscripcion.status FROM inscripcion, materia WHERE materia.clave = inscripcion.clave AND expediente =(?) AND materia.descripcion = 'ACTIVIDADES CULTURALES Y DEPORTIVAS' GROUP BY materia.descripcion UNION SELECT materia.descripcion, inscripcion.status FROM inscripcion, materia WHERE materia.clave = inscripcion.clave AND expediente =(?) AND materia.descripcion = 'PRÁCTICAS PROFESIONALES'; </pre>	<p>Se muestra si el alumno ha acreditado tanto sus actividades culturales y deportivas, así como sus prácticas profesionales.</p>
<pre> SELECT MIN(periodoMateria) AS minimo, MAX(periodoMateria) AS maximo FROM (SELECT COUNT(i.periodo) AS periodoMateria FROM inscripción i, alum_acad aa WHERE i.expediente = ? AND i.prog = aa.programa AND aa.expediente = ? GROUP BY i.periodo) AS per; </pre>	<p>Muestra el mínimo y el máximo de materias que ha llevado un alumno en los semestres cursados anteriormente.</p>
<pre> CREATE TABLE IF NOT EXISTS ? _ ?(id INT NOT NULL AUTO_INCREMENT, expediente INT NOT NULL, clave INT NOT NULL, descripcion VARCHAR(255) NOT NULL, campus VARCHAR(255) NOT NULL, </pre>	<p>Si no existe, se creará una tabla llamada programa_plan, con los campos id, expediente, clave de la materia, nombre de la materia, campus y periodo.</p>

periodo INT NOT NULL, PRIMARY KEY(id)) ENGINE = InnoDB;	
SELECT * FROM '?_?' WHERE expediente = ?;	Muestra si el alumno con el expediente ingresado ya ha solicitado materias en la tabla programa_plan (esto depende de su programa y plan del alumno).
DELETE FROM ?_? WHERE expediente = ?;	Elimina las materias solicitadas anteriormente por el alumno.
INSERT INTO '?_?' (expediente, clave, descripcion, campus, periodo) VALUES (?, ?, ?, ?, ?);	Se hace un registro donde se ingresan las materias seleccionadas por el alumno.

QUERYS

COORDINADORES

Query	Descripción
SELECT table_name FROM information_schema.tables WHERE table_name LIKE "" + eleccionPrograma + "%'	Nos regresa todas las posibles tablas de un programa, independientemente del plan.
SELECT DISTINCT(periodo) FROM ?;	Nos permite obtener los periodos disponibles para consultar en una de las tablas de solicitudes.
SELECT al.nombre,pr.expediente, pr.clave, pr.descripcion, pr.campus, pr.periodo, al.riesgo, al.riesgoant FROM pr, alumno al WHERE pr.expediente = al.expediente AND pr.campus = ? AND pr.periodo = ? ORDER BY riesgo DESC;	Nos envía la información general de las tablas, con los campos correspondientes, tomando en cuenta el campus y el periodo, si hay varios planes, se hace un UNION para juntar la información.
SELECT periodo, descripcion, COUNT(*) AS registros, SUM(CASE WHEN riesgo > 0 THEN 1 ELSE 0 END) AS en_riesgo, SUM(CASE WHEN riesgoant > 0 THEN 1 ELSE 0 END) AS en_riesgoant, CASE WHEN COUNT(*) BETWEEN 0 AND 29 THEN 'Poca demanda' WHEN COUNT(*) BETWEEN 30 AND 40 THEN 'Demanda suficiente' ELSE 'Mucha demanda' END AS demanda_abrir	Es una tabla resumiendo lo anterior, demostrando la cantidad de demanda que tiene, además de si es mucha demanda o no y cuantos alumnos solicitaron la materia están en riesgo o tuvieron algún riesgo, en la ultima subconsulta del SELECT, antes de terminar en el paréntesis, se le va concatenando un UNION en el programa dependiendo de cuantos planes del programa hayan solicitado materias.

<pre> FROM (SELECT pr.periodo, pr.descripcion, al.riesgo, al.riesgoant FROM programa_plan pr, alumno al WHERE pr.expediente = al.expediente AND pr.campus = ? AND pr.periodo = ?) AS tabla GROUP BY descripcion </pre>	
<pre> SELECT al.nombre,pr.expediente, pr.clave, pr.descripcion, pr.campus, pr.periodo, al.riesgo, al.riesgoant FROM ? pr, alumno al WHERE pr.expediente = al.expediente AND pr.campus = ? AND pr.periodo = ?; </pre>	<p>Nos muestra la misma información que la primera tabla pero ordenada por planes y no concatenadas una a la otra.</p>