

## ENTREGAS SPRINT 02 - GRUPO 10 EQUIPO 14

En el presente documento se adjuntan los avances relacionados con las tareas propuestas para el sprint 02 del proyecto “Teayudamos”, donde se encontrará lo siguiente:

- Maquetado completo de las siguientes páginas:
  - Landing page
  - Login page
- Modelado de la base de datos para el proyecto
- Scripts sql de creación de cada tabla.
- Propuesta de endpoints y su respectiva estructura de peticiones.
- Próximas tareas a desarrollar para el sprint 03.

La tareas aquí enumeradas hacen parte de la planeación hecha para el sprint 02 en nuestra tablero de Trello: <https://trello.com/b/ByWak9y2/avances-ciclo-3-grupo-14>

El informe de retrospectiva de los dos sprints se encuentra en el documento:

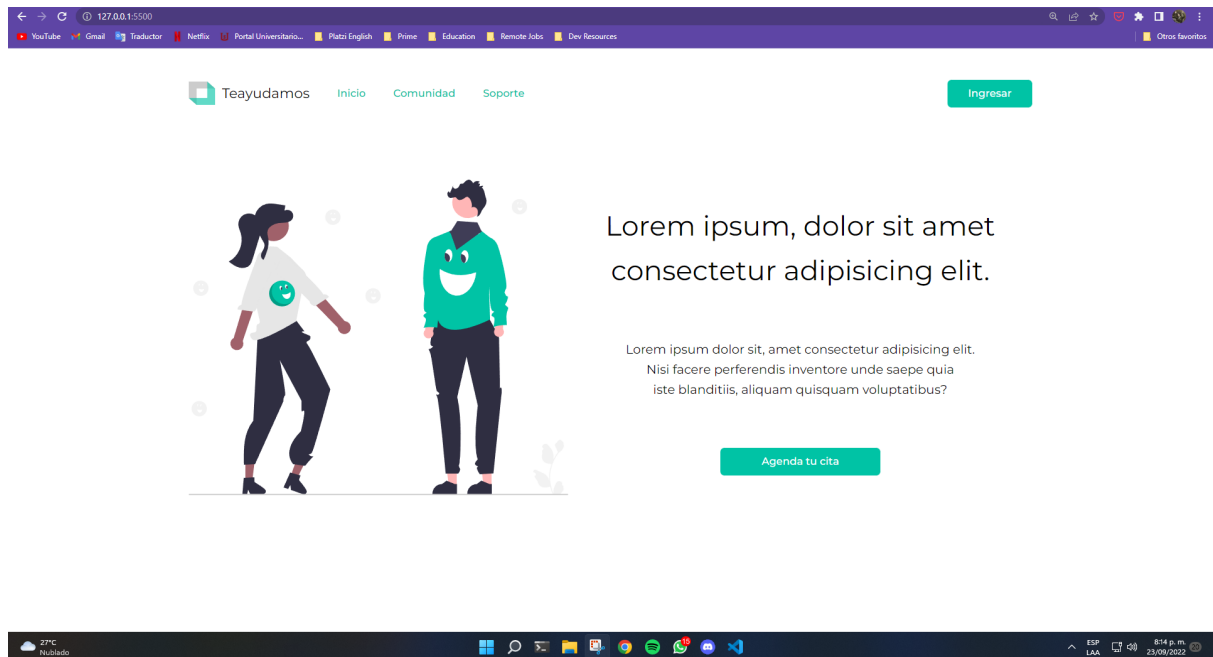
[https://docs.google.com/document/d/1I6JQwFpEvAgiHzMMA5c07N9Qculs1cVVXgWdn\\_-Ssmo/edit?usp=sharing](https://docs.google.com/document/d/1I6JQwFpEvAgiHzMMA5c07N9Qculs1cVVXgWdn_-Ssmo/edit?usp=sharing)

Y el contenido del proyecto, se encuentra almacenado en el repositorio de github:

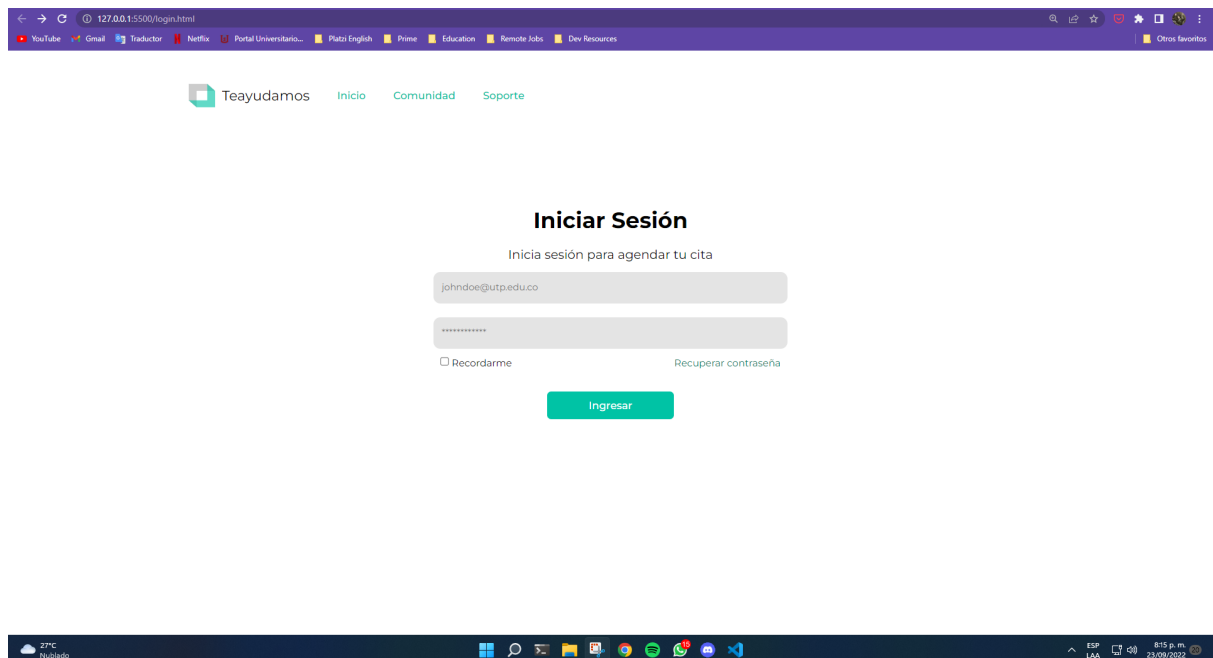
<https://github.com/JuanDiegoLA2998/Proyecto-Mision-Tic-ciclo-3>

# 01. Maquetado de páginas

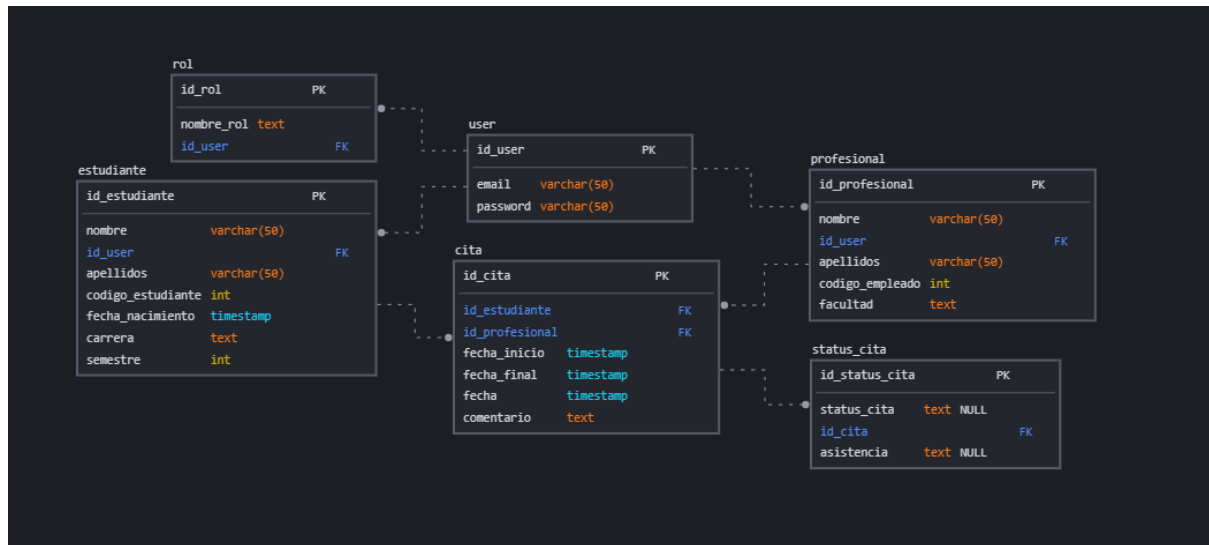
– Landing Page –



– Login page –



## 02. Modelado de la base de datos.



### 03. Scripts de creación de cada tabla.

```
CREATE TABLE cita
(
  id_cita          NOT NULL,
  id_estudiante    NOT NULL,
  id_profesional   NOT NULL,
  fecha_inicio     timestamp NOT NULL,
  fecha_final      timestamp NOT NULL,
  fecha            timestamp NOT NULL,
  comentario       text NOT NULL,
  CONSTRAINT PK_1 PRIMARY KEY ( id_cita ),
  CONSTRAINT FK_1 FOREIGN KEY ( id_estudiante ) REFERENCES
estudiante ( id_estudiante ),
  CONSTRAINT FK_2 FOREIGN KEY ( id_profesional ) REFERENCES
profesional ( id_profesional )
);

CREATE INDEX FK_2 ON cita
(
  id_estudiante
);

CREATE INDEX FK_3 ON cita
(
  id_profesional
);

CREATE TABLE estudiante
(
  id_estudiante    NOT NULL,
  nombre           varchar(50) NOT NULL,
  id_user          NOT NULL,
  apellidos        varchar(50) NOT NULL,
  codigo_estudiante int NOT NULL,
  fecha_nacimiento timestamp NOT NULL,
  carrera          text NOT NULL,
  semestre         int NOT NULL,
  CONSTRAINT PK_1 PRIMARY KEY ( id_estudiante ),
  CONSTRAINT FK_5 FOREIGN KEY ( id_user ) REFERENCES "user" ( id_user )
);

CREATE INDEX FK_2 ON estudiante
(
  id_user
);

CREATE TABLE profesional
(
  id_profesional   NOT NULL,
  nombre           varchar(50) NOT NULL,
  id_user          NOT NULL,
  apellidos        varchar(50) NOT NULL,
  codigo_empleado  int NOT NULL,
  facultad         text NOT NULL,
  CONSTRAINT PK_1 PRIMARY KEY ( id_profesional ),
  CONSTRAINT FK_4 FOREIGN KEY ( id_user ) REFERENCES "user" ( id_user )
);

CREATE INDEX FK_2 ON profesional
(
  id_user
);
```

```
CREATE TABLE rol
(
  id_rol          NOT NULL,
  nombre_rol      text NOT NULL,
  id_user         NOT NULL,
  CONSTRAINT PK_1 PRIMARY KEY ( id_rol ),
  CONSTRAINT FK_6_1 FOREIGN KEY ( id_user ) REFERENCES "user"
( id_user )
);

CREATE INDEX FK_2 ON rol
(
  id_user
);

CREATE TABLE status_cita
(
  id_status_cita   NOT NULL,
  status_cita      text NOT NULL,
  id_cita          NOT NULL,
  asistencia        text NOT NULL,
  CONSTRAINT PK_1 PRIMARY KEY ( id_status_cita ),
  CONSTRAINT FK_6 FOREIGN KEY ( id_cita ) REFERENCES cita ( id_cita )
);

CREATE INDEX FK_2 ON status_cita
(
  id_cita
);

CREATE TABLE "user"
(
  id_user          NOT NULL,
  email            varchar(50) NOT NULL,
  password         varchar(50) NOT NULL,
  CONSTRAINT PK_1 PRIMARY KEY ( id_user )
);
```

## 04. Endpoints propuestos

<b>Citas</b> Gestión de citas		^
POST	/crearCita Registro de nueva cita	✓ 🔒
GET	/nuevaCita/{profesional} Obtener disponibilidad del profesional	✓ 🔒
GET	/citaPendiente Obtiene cita pendiente del usuario	✓ 🔒
GET	/obtenerCitas Obtener historial de citas	✓ 🔒
GET	/obtenerCita/{cita} Obtener información de determinada cita	✓ 🔒
PUT	/modificarCita/{cita} Modifica cita con determinado ID.	✓ 🔒
PUT	/cancelarCita/{cita} Cancela cita con determinado ID.	✓ 🔒
<b>user</b> Operaciones directas del usuario		^
GET	/user/login Ingreso de usuario al sistema	✓
GET	/user/logout Desconecta al usuario del sistema	✓
GET	/user/{username} Obtiene a un usuario por su ID.	✓
PUT	/user/{username} Actualiza datos del usuario.	✓

## 05. Tareas próximo sprint

Para el próximo sprint esperamos desarrollar lo siguiente:

- Desarrollo de los endpoints de login, visualización y creación de citas.
- Consumo de API mediante Insomnia y documentación de cada prueba
- Desarrollo de las historias de usuario, para el siguiente sprint se complementará el trabajo realizado en el diseño de interfaces y navegación de todas las historias de usuario completamente, para esto el tester del equipo ha suministrado material y asesorías para facilitar trabajo en equipo, la armonía de este y solidificar las bases obtenidas durante las sesiones de clase, terminaremos de realizar la maquetación de las vistas faltantes y posible desarrollo herramientas y API's que faciliten el uso de la aplicación por parte de los clientes.