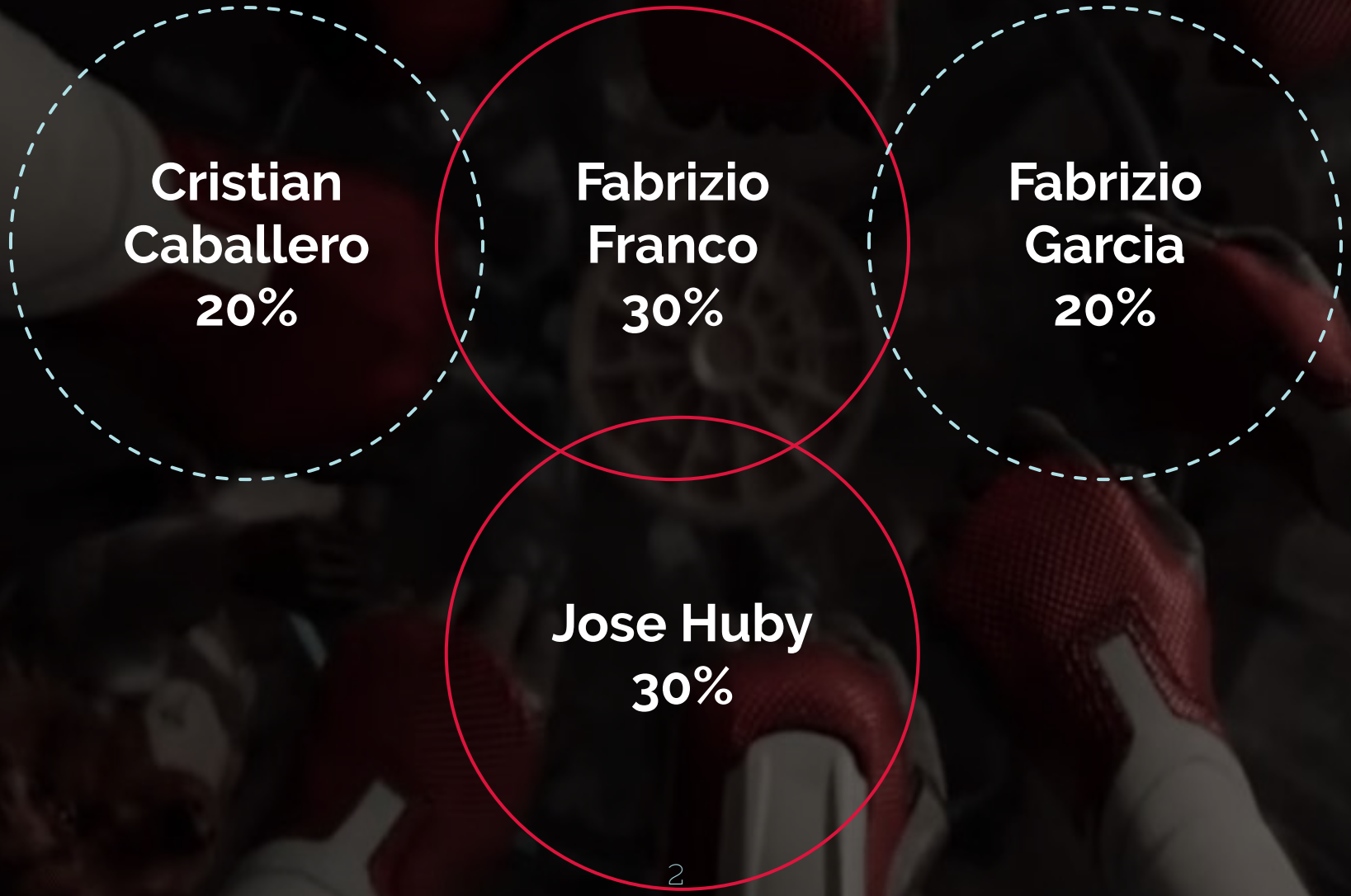
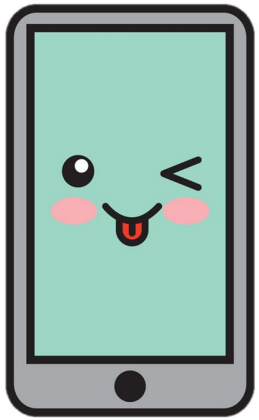
Entendí la referencia

INTEGRANTES



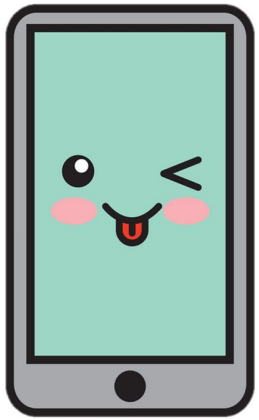
Acceso al aplicativo

https://bit.ly/2WLPDRL



Acceso al repositorio

https://bit.ly/2KsbVGn



Contexto

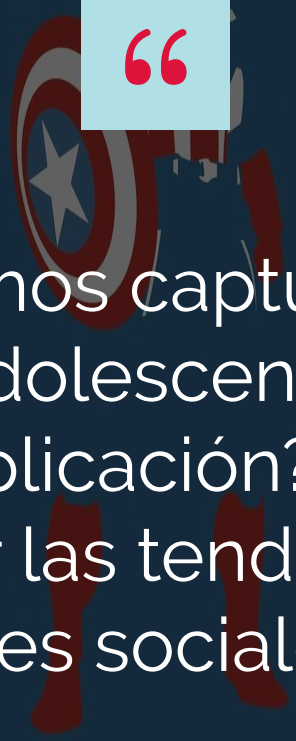
Problema:

Aburrimiento en tiempo libre

Necesidad:

Relajarse y divertirse

Propuesta



“¿Cómo logramos capturar la atención de miles de adolescentes estresados con una aplicación? … ¿y si nos guiamos por las tendencias de las redes sociales?”

SOLUCIÓN:

ENTENDÍ ESA REFERENCIA!

¿Qué es?:

Una aplicación web de entretenimiento dirigida a adolescentes en donde se deberá responder preguntas aleatorias ¡Se desafiará tu conocimiento sobre películas, ánimes, deporte y memes!

Objetivo: **Entretener**

Medio: Incentivos

Logramos

mediante

COMODIDAD
SIMPLEZA

Logramos

mediante

EMOCIÓN
COMPETENCIA

Logramos

mediante

ATENCIÓN OBSTÁCULOS

Logramos

mediante

GRATIFICAR PREMIOS

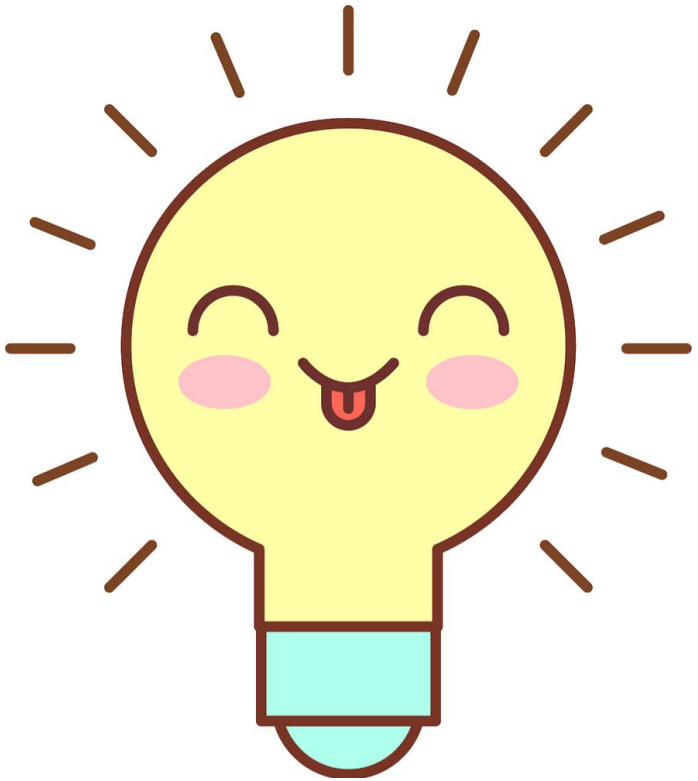
Mecánicas

REGISTRARSE

¡Únete en cuestión de segundos!

JUGAR

¡Gira la ruleta, obten una categoría al azar, responde la pregunta presentada y obtén puntos!



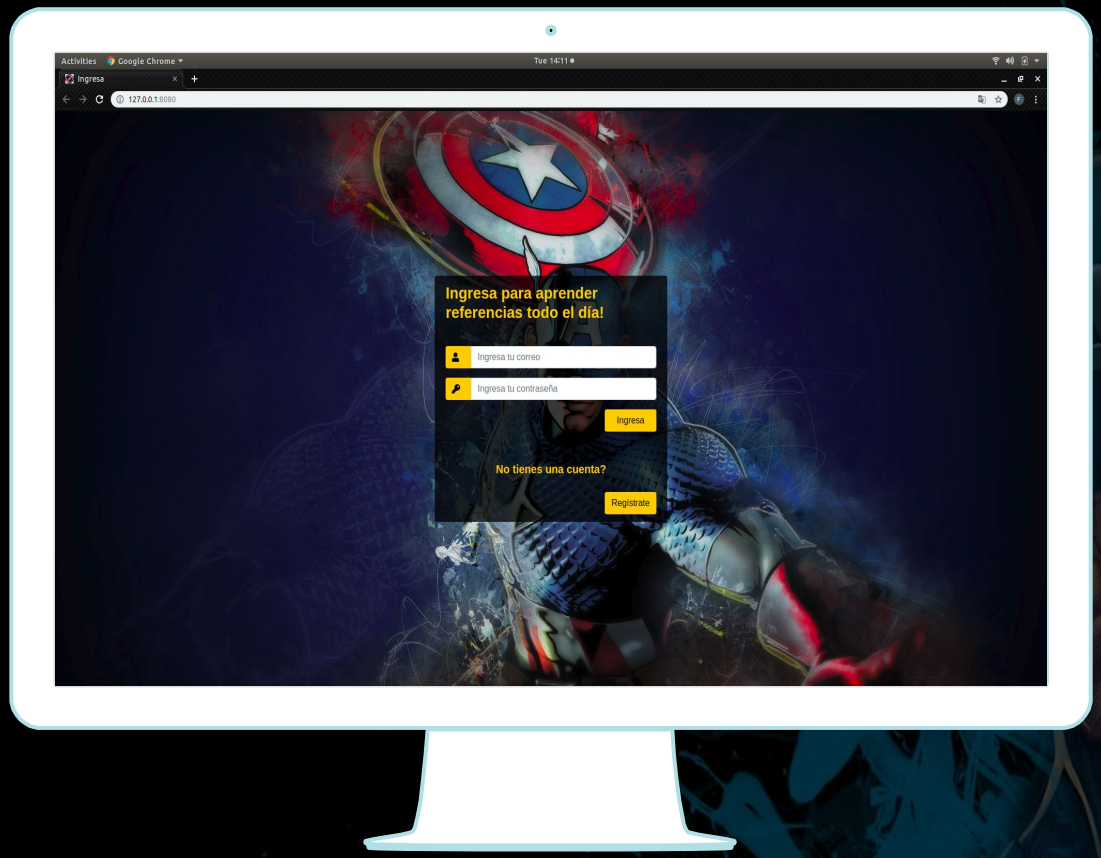
CREAR

¿Se te ocurrió algo? ¡Sube tu pregunta en cualquiera de nuestras categorías!

PRESUMIR

¡Hora de ser orgulloso! ¡Muestra tu puesto en base a tu puntaje!

Prototipo



Project demostration

Una pequeña demostración de lo que es nuestra plataforma web

https://bit.ly/2WLPDRL

Componentes esenciales

	**FLASK:** Administrar y entregar el contenido de nuestro servidor que solicita el usuario
	**jQuery:** Simplificar la programación en Javascript para plataformas web
	**AJAX:** Pedir información al servidor sin refrescar la página
	**BOOTSTRAP:** Proveer diseño amigable y minimalista
	**CANVAS:** Dibujar y animar contenido con facilidad

```
function siguiente_pregunta() {
  let id_ = JSON.stringify( value: { "id": (i+1) });
  $.ajax({
    url: "/set_category_and_random_question",
    type: "POST",
    contentType: "application/json",
    data: id_,
    dataType: "json",
      success: function(response) {
      establecer_pregunta(response);
    },
    error: function(response) {
      con_pregunta = false;
    }
  });
```

```
function shuffleArray(array) {
  for (let i = array.length - 1; i > 0; i--) {
    const j = Math.floor( x: Math.random() * (i + 1));
    [array[i], array[j]] = [array[j], array[i]];
  }
}

function establecer_pregunta(r) {
  statment = r.statment;
  respuestas_texto[0] = r.answer;
  respuestas_texto[1] = r.wrong1;
  respuestas_texto[2] = r.wrong2;
  respuestas_texto[3] = r.wrong3;
  shuffleArray(respuesta_orden);
}
```

```
@app.route('/set_category_and_random_question', methods = ['POST'])
def set_category_question():
    message = json.loads(request.data)
    id = message['id']
    db_session = db.getSession(engine)
    try:
        category = db_session.query(entities.Category
            ).filter(entities.Category.id == id
            ).one()
        category_id = category.id
        data = []
        categoryX=db_session.query(entities.Question).filter(entities.Question.category_id==category_id)
        for category in categoryX:
            data.append(category)
        randomx=random.choice(data)
        return Response(json.dumps(randomx, cls=connector.AlchemyEncoder), mimetype='application/json')
    except Exception:
        message = {'message': 'Unauthorized'}
        return Response(message, status=401, mimetype='application/json')
```

```
function Enviar(){
        $.ajax({
            url: 'http://3.130.127.150/questions',
            type: 'POST',
            contentType: 'application/json',
            data: JSON.stringify( value: {
                "statment": $('#statment').val(),
                "answer":$('#answer').val(),
                "wrong1":$('#wrong1').val(),
                "wrong2":$('#wrong2').val(),
                "wrong3":$('#wrong3').val(),
                "category_id":category_id
            }),
            dataType: 'json',
            success: function(response){
                AumentoUploads();
            window.location.href="http://3.130.127.150/gracias"
            }
```

```
@app.route('/questions', methods = ['POST'])
def create_question():
    sessiondb = db.getSession(engine)
    c =  json.loads(request.data)
    try:
        user = entities.Question(
        statment=c['statment'],
        answer=c['answer'],
        wrong1=c['wrong1'],
        wrong2=c['wrong2'],
        wrong3=c['wrong3'],
        category_id=c['category_id'])
        sessiondb.add(user)
        sessiondb.commit()
        message = {'message': 'Authorized'}
        return Response(message, status=200, mimetype='application/json')
    except Exception:
        message = {'message': 'Unauthorized'}
        return Response(message, status=401, mimetype='application/json')
```

Conclusiones: metodología de trabajo

- Es mejor trabajar en conjunto a trabajar en paralelo. (Google meets)
- Sabotea tu propio código para encontrar desperfectos.
- Testea la aplicación para agregar funcionalidades.
- Siempre es necesario revisar constantemente el código para evitar errores en cadena.

Gracias

Preguntas?