Il paroliere

Obiettivo del progetto è la realizzazione di un'applicazione ispirata al gioco in scatola *Il Paroliere*.

Le regole del gioco

Ad una partita de *Il Paroliere* possono partecipare da 2 a 6 giocatori. Il gioco richiede l'uso di: i) 16 dadi, riportanti su ogni faccia, una lettera dell'alfabeto; ii) una griglia di dimensione 4x4 sulla quale i dadi, una volta lanciati, verranno disposti; iii) un dizionario, usato per verificare l'esistenza delle parole; e iv) una clessidra della durata di 3 minuti.
Lo schema delle lettere riportate su ogni dado è mostrato nella Tabella 1.

Tabella 1: Lo schema dei dadi

		Faccia					
		1	***2***	***3***	***4***	***5***	***6***
Dado	***1***	B	A	O	O	Qu	M
	2	U	T	E	S	L	P
	3	I	G	E	N	V	T
	4	O	U	L	I	E	R
	5	A	C	E	S	L	R
	6	R	A	T	I	B	L
	7	S	M	I	R	O	A
	8	I	S	E	E	F	H
	9	S	O	T	E	N	D
	10	A	I	C	O	F	R
	11	V	N	Z	D	A	E
	12	I	E	A	T	A	O
	13	O	T	U	C	E	N
	14	N	O	L	G	U	E
	15	D	C	M	P	A	E
	16	E	R	I	N	S	H

Lo scopo del gioco è di misurare l'abilità dei giocatori nel riconoscere parole componibili con le lettere mostrate dai dadi. Una partita è caratterizzata da un numero variabile di sessioni di gioco. Una sessione di gioco inizia con il lancio e la disposizione casuale dei 16 dadi sulla griglia 4x4. Un esempio di griglia è riportato in Figura 1, assieme alla disposizione dei dadi (da 1 a 16) che hanno generato tale schema di lettere.

a)

G	A	U	T
P	R	M	R
D	O	L	A
E	S	I	C

b)

3	1	4	2
15	5	7	6
11	9	14	12
16	8	10	13

Figura 1. Griglia di lettere (a), e disposizione dei dadi che l'hanno generata (b).

Una volta disposti i dadi viene girata la clessidra. L'obiettivo per i giocatori, è di identificare ed annotare, entro 3 minuti (la durata della clessidra), il maggior numero di parole, di lunghezza maggiore possibile, che possono essere composte concatenando le lettere indicate nella griglia. Più precisamente, le parole vengono composte partendo dalla lettera presente in una cella della griglia, e visitando le celle adiacenti, muovendosi verticalmente, orizzontalmente od obliquamente di una posizione in qualsiasi direzione, evitando di visitare la stessa cella più di una volta. Le parole

identificate sono considerate ammissibili solo se corrispondono a sostantivi o aggettivi nella forma singolare, o a verbi all'infinito, e se sono indicizzate nel dizionario, ovvero se corrispondono ad un vocabolo utilizzabile come chiave di ricerca. Esempi di parole ammissibili per la griglia in Figura 1 sono: DOSE, ORMA, TRAMA, PRODE ed ARDESIA. La parola SILOS non è ammissibile in quanto richiederebbe di visitare la cella contenente la S due volte, mentre la parola TRAUMA non è ammissibile in quanto composta da lettere presenti in celle non adiacenti tra loro. Infine, la parola MORDE non è ammissibile poiché è un verbo all'indicativo, mentre il dizionario indicizza i verbi all'infinito (ad es. mordere).
Al termine dei 3 minuti caratterizzanti la sessione di gioco, ogni giocatore comunica la lista di parole che ha identificato, e le parole vengono analizzate, in base ai criteri precedentemente citati, per verificarne l'ammissibilità. Le parole non ammissibili vengono scartate, così come le parole reputate ammissibili che sono state proposte da più giocatori. Per ogni parola rimanente, un giocatore riceve un quantitativo di punti:

- per ogni parola di 3 o 4 lettere: 1 punto
- per ogni parola di 5 lettere: 2 punti
- per ogni parola di 6 lettere: 3 punti
- per ogni parola di 7 lettere: 5 punti
- per ogni parola di 8 o più lettere: 11 punti

Vince la partita il giocatore che raggiunge per primo la soglia di 50 punti guadagnati.
Se in una singola sessione di gioco nessun giocatore riesce ad accumulare 50 punti, si tiene traccia dei punti guadagnati da ogni giocatore, e il gioco viene reiterato ripartendo con una nuova sessione. Le reiterazioni terminano quando il punteggio complessivo di un giocatore raggiunge la soglia.

Requisiti

L'applicazione *Il Paroliere (IP)* dovrà essere utilizzata da utenti con ruoli distinti: 1) il *giocatore*, che organizza e partecipa a delle partite; e 2) l'*amministratore*, che gestisce le componenti di backend della piattaforma.
La piattaforma consiste di: 1) un modulo *serverIP*, che interfacciandosi con un DBMS relazionale (PostgreSQL), e con un dizionario della lingua italiana, fornisce servizi di back-end; 2) un modulo *playerIP*, che fornisce servizi designati per i giocatori. IP dovrà essere realizzata in modo tale da supportare l'interazione in parallelo con più utenti connessi alla piattaforma da postazioni differenti. IP dovrà essere realizzata in Java utilizzando tecnologie che permettano un'implementazione efficiente dei servizi.

Avvio del server

All'invocazione di *serverIP* deve essere richiesto di specificare: i) le credenziali per accedere a *dbIP*, un database di supporto all'esecuzione dei servizi della piattaforma *IP*, e ii) l'host del database. Se *dbIP* non contiene un profilo di utente amministratore, *serverIP* deve avviare un processo di registrazione che richiederà all'utente di specificare un identificativo e una password. Le credenziali registrate (uid e pwd) potranno essere utilizzate nei successivi avvii del server per effettuare l'autenticazione dell'utente amministratore. Al termine della registrazione, o nel caso in cui in *dbIP* sia già presente un utente con profilo di amministratore, *serverIP* dovrà rimanere in attesa di richieste di connessione da parte di client *playerIP*.

Registrazione utenti

Un utente che intende accedere a *IP* deve necessariamente registrarsi.
La registrazione dei concorrenti è gestita da *playerIP*. Il processo di registrazione richiederà all'utente di fornire il nome, il cognome, un nickname, un indirizzo di posta elettronica e una password. L'indirizzo di posta elettronica e il nickname non devono essere già stati usati da altri utenti. Prima di completare la registrazione la piattaforma invierà all'utente un codice di attivazione

da immettere per finalizzare la registrazione. Se l'utente non completa la registrazione entro 10 minuti dalla spedizione del codice il profilo dovrà essere cancellato.

Autenticazione

L'accesso ad ogni funzionalità offerta dalla piattaforma tramite *playerIP* (ad eccezione del servizio di registrazione utenti e di ripristino password dimenticata) richiede l'autenticazione dell'utente. Un utente effettua il login specificando come identificativo l'indirizzo di posta e la password associata al proprio profilo. Un giocatore autenticato può:

1. organizzare una nuova partita,
2. visualizzare le partite organizzate per la quale non si sono ancora chiuse le iscrizioni dei concorrenti, e le partite già in corso di svolgimento,
3. richiedere la partecipazione, ad una partita organizzata da altri utenti per la quale non sono ancora state chiuse le iscrizioni, ed iniziare a giocare una volta terminate le iscrizioni
4. abbandonare, una partita per la quale non sono ancora state chiuse le iscrizioni, oppure che è in corso di svolgimento
5. modificare i dati del proprio profilo
6. analizzare le statistiche di utilizzo della piattaforma di gioco

Organizzare una nuova partita

Un utente autenticato, mediante *playerIP*, deve poter organizzare una nuova partita, stabilendone il nome, e il numero di giocatori che possono parteciparvi (da 2 a 6). L'utente organizzatore viene automaticamente iscritto alla partita organizzata.

Visualizzare la lista delle partite

Il client *playerIP* permette agli utenti autenticati di visualizzare l'elenco delle partite organizzate con iscrizioni ancora aperte, e di quelle in corso di svolgimento. Per ogni partita in elenco viene mostrato il nome, la data, e l'ora di creazione, il numero di giocatori previsti, il numero di quelli iscritti ed i rispettivi nickname.

Richiesta di partecipazione ad una partita

Un utente autenticato, mediante *playerIP* può richiedere di partecipare ad una partita organizzata da altri utenti con iscrizioni ancora aperte. *IP* deve autorizzare la partecipazione se il numero di giocatori già iscritti è inferiore al numero di giocatori previsti per la partita dall'organizzatore.

Svolgimento di una partita

Al raggiungimento del numero di giocatori necessari a far iniziare una partita, *IP* chiude le iscrizioni e notifica l'inizio imminente del gioco ai giocatori registrati. I moduli *playerIP* dei giocatori iscritti visualizzano un timer che effettua un conto alla rovescia di 30 secondi, corrispondente al tempo mancante all'inizio effettivo del gioco (i timer dei giocatori devono essere sincronizzati). Allo scadere del timer inizia l'effettiva sessione di gioco. All'inizio della sessione *IP* simula: i) il lancio dei 16 dadi, generando casualmente la lettera mostrata da ogni dado, e la disposizione casuale dei dadi nella griglia, e ii) il ribaltamento della clessidra, facendo partire un timer che effettua un conto alla rovescia di 3 minuti. Il modulo *playerIP* dei giocatori dovrà visualizzare:

1. Il nome della partita in corso di svolgimento, e l'ordinale della sessione di gioco avviata
2. la griglia 4x4 riportante le lettere selezionate
3. il timer che indica i minuti e secondi mancanti al completamento della sessione di gioco
4. i nomi e i punteggi di tutti i giocatori partecipanti

Inoltre, *playerIP* deve permettere al giocatore di annotare e visualizzare le parole che ha identificato nella sessione di gioco (le parole annotate dagli altri giocatori non sono visibili).

Allo scadere del timer, *IP* dovrà analizzare le parole annotate dai vari giocatori, verificando, per ogni parola, che sia componibile visitando la griglia in base ai criteri di navigazione precedentemente considerati, che sia di lunghezza almeno pari a 3 lettere, che sia un vocabolo presente nel dizionario, e che non sia stata proposta da più giocatori. *playerIP* dovrà mostrare ad ogni giocatore la lista delle parole identificate da ognuno, il punteggio apportato da ogni parola, il punteggio acquisito nella sessione di gioco, il punteggio cumulativo dell'intera partita per ogni giocatore. Nel caso di parole con punteggio associato pari a 0, dovrà essere indicata la ragione del mancato apporto di punti, ovvero se la parola non è derivabile dalla griglia, non è presente nel dizionario, è di lunghezza insufficiente, o è stata indicata da altri (oppure una combinazione di questi criteri). Il giocatore dovrà poter verificare la definizione di qualsiasi parola appartenente alla lista delle parole trovate dagli altri giocatori o eventualmente anche alla propria lista. Più precisamente, *playerIP* dovrà permettere al giocatore di selezionare le parole che intende verificare, e dovrà mostrare la loro definizione derivata dal dizionario. Ogni giocatore avrà fino a 3 minuti di tempo per verificare le parole. *playerIP* dovrà mostrare un timer che indica il tempo rimanente, ed un pulsante che dovrà permettere al giocatore di notificare ad *IP* di aver terminato la verifica. Il turno di analisi termina quando tutti i giocatori iscritti hanno notificato di aver completato l'analisi, oppure, se tale condizione non si verifica, alla scadenza del timer. Una volta completata l'analisi, se il punteggio cumulativo di almeno un giocatore raggiunge la soglia di 50 punti, *playerIP* dovrà mostrare il vincitore della partita. In caso contrario, *IP* dovrà riproporre una nuova sessione di gioco.

Abbandono di una partita

playerIP deve permettere ai giocatori di abbandonare la partita a cui stanno partecipando. Se un giocatore abbandona la partita in corso di svolgimento, la partita viene annullata, tutti i giocatori ricevono notifica dell'annullamento, e la partita viene rimossa dalla lista delle partite organizzate. Se invece l'abbandono avviene prima della chiusura delle iscrizioni, il giocatore viene rimosso dalla lista dei concorrenti iscritti. Se in seguito all'abbandono la lista dei concorrenti iscritti è vuota, la partita viene annullata e rimossa dalla lista delle partite organizzate. *IP* dovrà tener traccia della partecipazione ad ogni sessione di gioco, delle parole identificate e dei relativi punteggi acquisiti.

Modifica dei dati di profilo

Un utente, mediante *playerIP* deve poter modificare i dati del proprio profilo.
In particolare, l'utente potrà correggere nome e cognome, e cambiare nickname, e password. L'impostazione di una nuova password è soggetta all'inserimento della vecchia password. IP dovrà inviare una mail all'indirizzo di posta registrato informandolo del cambio avvenuto (senza mostrare le password).

Reset di password

playerIP deve fornire un servizio per reimpostare una password dimenticata, accessibile anche da utenti non autenticati. Il servizio deve richiedere all'utente di specificare l'email associata al profilo, e dovrà rimpiazzare la password del profilo con una nuova generata automaticamente, inoltrandola all'indirizzo di posta elettronica associato al profilo.

Monitoraggio

playerIP deve offrire servizi di analisi. In particolare deve:

1. Mostrare il giocatore:
    - che detiene il primato di punteggio raggiunto per sessione e per partita, indicando i rispettivi punteggi;
    - che ha giocato più sessioni di gioco in assoluto, indicandone il numero;

    - con la media più alta di punti acquisiti per sessione di gioco e per partita, indicando i rispettivi punteggi;
    - che ha proposto il più alto numero di parole duplicate, ovvero indicate da almeno un altro giocatore nella stessa sessione di gioco, indicandone il numero;
    - che ha proposto il più alto numero di parole che non possono essere derivate dalla configurazione dei dadi mostrata sulla griglia, oppure che non sono presenti nel dizionario, indicandone il numero,
2. Mostrare la classifica delle occorrenze delle parole valide che sono state correttamente identificate.
3. Mostrare le parole la cui identificazione ha causato il più alto incremento di punti, ed indicare l'incremento corrispondente, riferendo, per ognuna, la partita in cui è stata proposta
4. Mostrare il numero medio di turni di gioco che caratterizzano una partita interamente giocata, da gruppi di 2, 3, 4, 5 e 6 giocatori
5. Mostrare il numero minimo e massimo di turni che caratterizzano una partita interamente giocata da gruppi di 2, 3, 4, 5 e 6 giocatori
6. Derivare l'occorrenza media delle lettere mostrate apparse sulla griglia nei vari turni di gioco
7. Mostrare le parole di cui è stata richiesta almeno una volta la definizione, ordinandole dalla più richiesta alla meno richiesta.
8. Mostrare l'identificativo delle partite in cui è stata richiesta la definizione di una parola data.

Requisiti dati

serverIP fornisce servizi di supporto alla memorizzazione ed analisi dei dati dell'intera piattaforma di gioco. E'richiesto di progettare la base di dati che *serverIP* utilizzerà per gestire informazioni relative agli utenti, e alle partite giocate. In particolare, sono di interesse le seguenti informazioni:

- il profilo dell'amministratore, e dei giocatori regolarmente registrati, caratterizzato dal cognome, il nome, un nickname, un indirizzo email, e una password.
- lo storico dei turni di gioco e delle partite che sono state giocate e che si stanno giocando, tenendo traccia: delle lettere mostrate sulla griglia, delle parole identificate dai giocatori, dei punti acquisiti con ogni parola, e delle richieste di definizione di parole.

Svolgimento del progetto

L'obiettivo del progetto consiste nella 1) progettazione e 2) sviluppo di un sistema software e della relativa base di dati che soddisfi i requisiti proposti, 3) utilizzando un processo di sviluppo object oriented strutturato, e 4) nella produzione di artefatti (quali, diagrammi UML e schemi ER) che descrivano quanto realizzato nelle varie fasi dello sviluppo, in conformità alle metodologie seguite a lezione.
Le attività di analisi e progettazione devono essere adeguatamente documentate facendo uso del linguaggio UML per l'applicazione software e del modello Entity-Relationship (ER) per il database.
È richiesto di progettare l'applicazione avvalendosi dove possibile dell'uso di design patterns, e di realizzare l'applicazione con un'opportuna interfaccia grafica, usando il linguaggio di programmazione Java, e progettare e realizzare un database utilizzando PostgreSQL per la sua implementazione (http://www.postgresql.org) e JDBC per l'accesso alla base di dati da programma Java (http://jdbc.postgresql.org/download.html)
È opportuno evidenziare che i servizi dell'applicazione vengono erogati in parallelo a più utenti, e che possono verificarsi accessi concorrenti a risorse condivise. L'applicazione deve quindi essere definita in modo tale da gestire opportunamente l'accesso concorrente alle risorse accedute.

Realizzazione del Database

La realizzazione del database prevede svariate fasi. In seguito si riportano le attività ed i relativi artefatti che dovranno essere prodotti nelle varie fasi.

- Si ristrutturi, se necessario, secondo le metodologie di progettazione i requisiti descritti. Si scelgano le metodologie per la costruzione dello schema ER, motivando le scelte fatte.
- Si definisca lo schema concettuale ER per il database, evidenziando le entità e le associazioni di interesse, nonché i vincoli di cardinalità e di identificazione, motivando le scelte effettuate. Altri eventuali vincoli devono essere espressi in linguaggio naturale.
- Si effettui la ristrutturazione dello schema ER motivando le scelte effettuate.

E' richiesto di produrre un documento di analisi dei requisiti ristrutturato e documentazione associata allo schema ER (ristrutturato e non), con eventuale specifica di vincoli in linguaggio naturale.

- Si effettui la traduzione dello schema ER ristrutturato in un equivalente schema relazionale.

E' richiesto di produrre la documentazione associata allo schema relazionale derivato dallo schema concettuale.

- Si realizzi il database utilizzando PostgreSQL, e SQL per la definizione dei dati, l'implementazione dei vincoli identificati, e la manipolazione dei dati, secondo le operazioni previste dall'applicazione.

Documentare gli script SQL necessari alla creazione della base di dati e dei vincoli definiti sui dati e le query SQL a supporto dei servizi erogati da IP.

Codifica

Il codice sorgente dell'applicazione dovrà essere definito in modo da facilitarne la comprensione e la manutenibilità. Per favorire questo obiettivo vengono in seguito riportate alcune linee guida:

- Ogni volta che si identificano due estratti di codice sufficientemente simili, il codice deve essere reingegnerizzato, e il codice duplicato eliminato portando a fattore comune le funzionalità condivise. Si consiglia l'utilizzo di strumenti di refactoring forniti da ambienti di sviluppo quali, ad esempio, Eclipse / NetBeans.
- E' opportuno considerare che esistono delle "regole di stile" per ogni linguaggio di programmazione, quindi per indentazione, nomi, parentesi, commenti, etc.

Documentazione del codice

Generare documentazione javadoc documentando i metodi e gli attributi con visibilità *public* o *protected* di ogni classe con visibilità *public*. La documentazione generata deve spiegare come usare una determinata classe o interfaccia, non come questa sia stata effettivamente implementata, se questo può essere tenuto nascosto.
Prima di commentare il codice perché altrimenti incomprensibile, tentare le seguenti strade:

- Se si sono usati nomi poco (o del tutto non) chiari, sostituirli con nomi che spieghino lo scopo dell'entità (variabile locale, parametro, attributo, metodo, classe, package...)
- Se il corpo di un metodo è troppo grande, scomporre il metodo in più metodi (eventualmente con visibilità private)
- Se i metodi o gli attributi di una classe sono troppo numerosi, scomporre la classe in classi con meno responsabilità (eventualmente con visibilità package)
- Se la classe svolge uno specifico ruolo di un pattern, specificare il ruolo, il pattern e gli altri partecipanti (quali altri classi e che ruolo hanno); non descrivere il funzionamento del pattern.
- Se il codice rimane comunque di difficile comprensione, inserire commenti in linguaggio naturale.

Aiuti implementativi

IP per effettuare l'analisi della validità delle parole e per vederne il significato si avvale del dizionario della lingua italiana utilizzato da LibreOffice (https://extensions.libreoffice.org/extensions/italian-dictionary-thesaurus-hyphenation-patterns).

E’ necessario scaricare il file *dict-it.oxt* contenente il dizionario. In allegato alla presente documentazione viene fornito il codice sorgente di alcune classi Java che esemplificano come possa essere effettuata la ricerca di un vocabolo nel suddetto dizionario (la classe principale è Loader) interagendo con il file scaricato. Per gestire l’interazione è suggerito di ispirarsi a tale codice, adattandolo alle esigenze di IP.

Come descritto, l'interazione tra IP e l'utente finale può avvenire anche per mezzo di email composte e inoltrate automaticamente. La realizzazione di tale funzionalità richiede l'utilizzo di servizi di inoltro di messaggi di posta elettronica forniti da un server SMTP. JavaMail (https://java.net/projects/javamail), è una libreria che supporta l'interfacciamento di applicazioni Java con vari servizi di gestione della posta elettronica, tra cui quelli forniti da server SMTP. E' pertanto richiesto di utilizzare JavaMail (https://maven.java.net/content/repositories/releases/com/sun/mail/javax.mail/1.5.6/javax.mail-1.5.6.jar) per supportare l'interazione tra IP e un server SMTP. In allegato alla presente documentazione viene fornito il codice sorgente di una classe che effettua la spedizione di una mail usando il server SMTP usato in ateneo (smtp.office365.com). Per implementare la spedizione di email è suggerito di ispirarsi a tale codice, riutilizzandolo, ed eventualmente modificandolo. Il server SMTP considerato consente l'invio di email esclusivamente ad utenti autenticati. Le credenziali di autenticazione sono le stesse usate per il login al servizio di web mail dell'università (ad es. username: nome.cognome@studenti.uninsubria.it pwd: mypassword). La spedizione è autorizzata solo se le credenziali sono valide e il mittente del messaggio corrisponde allo username.

Aiuto nella preparazione della documentazione

Per facilitare la preparazione della relazione, in allegato viene fornito un documento riportante la struttura del documento da consegnare, con l’indicazione degli artefatti che è necessario presentare.

Suddivisione del lavoro

Si richiede di indicare:

- come è stato suddiviso il prodotto software. Ad es. il programma è stato diviso in “moduli”: GUI, gestione/elaborazione dati, etc.
- come è stato organizzato il processo di sviluppo.
- A chi sono state allocate le diverse attività rispetto alle varie parti del prodotto. Ad es. l’analisi è stata fatta da X, la progettazione da X e Y in coppia, etc.

Valutazione del progetto

È auspicabile che il progetto possa essere un'occasione formativa di lavoro in team. Gli studenti sono pertanto invitati a svolgere il progetto in gruppi di lavoro composti da 3 / 4 studenti.
La valutazione del progetto avviene singolarmente per ciascuno studente. Il giorno dell’appello verrà comunicato tramite avviso sulla pagina del corso nella piattaforma di elearning. Durante la valutazione è richiesto allo studente di saper argomentare in modo opportuno le scelte progettuali, algoritmiche, e implementative adottate. In fase di discussione orale verrà verificata l'effettiva padronanza delle tecniche utilizzate attraverso una serie di domande.
La valutazione terrà conto dei seguenti fattori: l’aderenza del sistema realizzato ai requisiti proposti, i documenti di analisi e progettazione prodotti sia per la realizzazione del software che per il database (correttezza sintattica, semantica, completezza e leggibilità, minimalità dello schema logico), le scelte algoritmiche e di progettazione effettuate (design pattern), la qualità del codice sorgente prodotto (funzionalità, correttezza, facilità d’uso).

Consegna

Il progetto deve essere consegnato entro il giorno stabilito per ogni appello d'esame, utilizzando l'apposita form disponibile sulle pagine del Laboratorio Interdisciplinare B della piattaforma di e-

learning. Alla voce commenti, specificare il nome e le matricole degli studenti del gruppo coinvolto. Il progetto deve essere allegato in formato compresso ZIP.
I progetti devono essere corredati da:

- documentazione
- codice sorgente
- file di build per Apache Ant (http://ant.apache.org/) o Maven (https://maven.apache.org/) per compilare il progetto, lanciare il server e i client, creare il database, creare la documentazione javadoc, etc.
- eventuali librerie necessarie alla compilazione e/o all'esecuzione
- file README con indicazioni sull'installazione e sulla compilazione, specificando i comandi Ant/Maven da utilizzare, ed indicazioni di particolari librerie, usate in modo non standard.

Il progetto deve compilare correttamente (tramite il file di build), una volta espanso in una directory.
I progetti con errori di compilazione e/o di esecuzione non verranno valutati.
Una volta effettuata la consegna, agli studenti verrà comunicata una data in cui potranno sostenere la discussione orale del progetto e la valutazione del lavoro svolto.

Domande sul Progetto

Per qualsiasi dubbio durante lo svolgimento del progetto: pietro.colombo@uninsubria.it , l'oggetto della mail deve essere: "Laboratorio B: Domande"
Progetto valido a partire dall'appello di Giugno 2020 a Febbraio 2021.